MODELLING AND HEURISTIC APPROACHES TO RETAIL SPACE ALLOCATION IN COMPLEX ENVIRONMENTS

A Thesis Submitted in

Partial Fulfillment of the Requirements

for the degree of

Master of Technology

by

Moholkar Makarand Prabhakar

to the

DEPARTMENT OF INDUSTRIAL AND MANAGEMENT ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

February 2001

CERTIFICATE

It is certified that the work contained in this thesis entitled "***Modelling and Heuristic Approaches to Retail Space Allocation in Complex Environments***" has been carried out by **Mr. Moholkar Makarand Prabhakar (Roll No. 9911410)** under my supervision and this work has not been submitted elsewhere for a degree.

February, 2001

(Sanjeev Swami)

Assistant Professor
Industrial and Management Engineering
Indian Institute of Technology
Kanpur-208016
India

ACKNOWLEDGEMENT

I wish to express my deep sense of gratitude and indebtedness towards Dr. Sanjeev Swami for his inspiring guidance, invaluable suggestions and constructive criticism. He was always a constant source of encouragement throughout my thesis work, and was more than a thesis supervisor.

I would also like to extend my sincere thanks to Ramkrishna, Garima, Punit, Rajkumar, Sagar, Vishwesh, Vinod and others for making my stay at IITK, a smooth and memorable one.

I extend my thanks to IME faculty members and staff for showing a special bondage towards me as an IME family member.

- *Makarand.*

CONTENTS

LIST OF FIGURES AND TABLES

ABSTRACT

Retailing is becoming an increasingly important area of management attention and academic research. The development of decision support models to help retailers improve their decision-making is gaining importance. Every week motion picture exhibitors have to make an important decision regarding the replacement of the movies playing at the screens in their theaters. The dynamic and uncertain decision environment, complicated contract terms, a number of new products and their perishing demand give rise to the complexity of the problem. In several instances, this complexity is further increased by availability and choice of mupltiple display facilities.

In this thesis, we develop modelling and heuristic methods to help the managers in this industry improve their decision making regarding the optimal use of their display space. A Markov decision process based model is used to address the problem of exhibitors with single screen. We propose a Genetic Algorithm based heuristic to address the similar problem with multiple screens. We investigate the improvement shown by our heuristic over other heuristics. We also discuss the extensions of an integer programming model SilverScreener, to address similar problems in other application areas.

CHAPTER 1

INTRODUCTION

Marketing is "the process of planning and executing the conception, pricing, promotion, and distribution of ideas, goods and services to create exchanges that satisfy individual and organizational objectives." (Kotler, 2000,P. 8). Often marketing systems are quite complex so that marketing researchers need to develop analytical models to manage these systems. There are two basic methodologies for modeling in marketing: verbal and mathematical. Most of the models in the so-called "behavioural literature" in marketing are cast as verbal models. For example, Howard and Sheth's (1969) theory of consumer behaviour is a verbal model of consumer behaviour. Mathematical models, popularly known as analytical or quantitative models, use symbols to denote marketing variables and express their relationship as equations or inequalities. The analysis follows the rules of mathematical logic. For example, Bass's (1969) model of product diffusion is one of the most frequently used quantitative models in marketing.

Quantitative modeling in the field of marketing is progressing rapidly. These promising analytic and databased findings are important to the development of a marketing theory that enables marketers to understand and manage markets more effectively in an increasingly dynamic and complex environment. In particular, there has been a rapid development in the research addressing the issues related to dynamic decision making. Examples can be found in areas such as advertising (Erickson 1995), brand choice (Erdem and Keane 1996), new product development (Bayus 1995), channel coordination (Chintagunta and Jain 1992), consumer behavior (Krishna 1994), market research (Bockholt and Dillon 1997), pricing (Raman and Chatterjee 1995), promotion (Papatla and Krishnamurthi 1996), and retailing (Krider and Weinberg 1997).

Retailing, one of the important functions of marketing, is becoming an increasingly important area of management attention and academic research, especially in the developed countries. One of the major areas of research in retailing is

the development of decision support models to help retailers improve their decision-making in the dynamic and complex environment. Most previous marketing research in this regard has been concerned with consumer durable goods, not with services or intangibles. This thesis seeks to advance our understanding of how quantitative model building can improve marketing decision making in complex dynamic environments by focusing on a perishable entertainment product, namely, movies. Though we focus on movies in this thesis, our methodology and results can readily be generalized to other entertainment products (e.g., performing arts, books, video games), travel services, fashion goods, and educational programs.

Shelf (display) space management is an area of important concern to retailers. First, consumers choose from the products that are displayed on the shelf. In this sense, shelf space management determines the ultimate profitability of the retailer. Second, most retailers have limited shelf space to display their products. Therefore, the choice of which products to display becomes an important retailing decision. The decision is particularly complex in those industries in which many new products are continually introduced in market. Finally, anticipating and adapting to dynamic changes in consumers' tastes and demand are key concerns to most retailers. Moreover, such decisions have to be made for several product categories. Though previous researchers (Corstjens and Doyle 1983; Bultez and Naert 1988; Borin, Farris, and Freeland 1994) have addressed some of these issues, they have generally been in the context of packaged goods in a supermarket chain setting. However, other industry settings, such as the motion picture (or movie) industry considered in this thesis, pose different challenges and intriguing problems.

A movie is an interesting product for several reasons. Most movies are separate entities and can be considered an innovation because of the new features (e.g., actors, storyline, music, etc.) usually included in them. These new movies are released every week throughout the year. The product life cycle of movies is relatively short and is measured in weeks. The product is seasonal in nature and the prominent seasons are during holidays. Finally, one of the most interesting attribute of movies from a research standpoint, and a major component of the complexity associated with the problems addressed in this thesis, is their perishability.

In response to the dynamics and challenges posed by the above characteristics of movies, a stream of research, particularly addressing the marketing of movies has contributed to the marketing literature. At the consumer behavior level, some of the research has questioned the relevance of the traditional information-seeking framework for studying the consumption of movies (e.g., Hirschman and Holbrook, 1982; Holbrook and Hirschman 1982). Another stream of research has focused on forecasting the enjoyment of movies at the individual level (Eliashberg and Sawhney 1994) as well as forecasting the commercial success of movies at the aggregate level (Smith and Smith 1986; Austin and Gordon 1987; Dodds and Holbrook 1988; Sawhney and Eliashberg 1996; Eliashberg and Shugan 1997). Additionally, some research has begun to emerge addressing diffusion (Mahajan, Muller, and Kerin 1984; Jones and Ritz 1991), seasonality (Radas and Shugan 1995), release timing (Krider and Weinberg 1998), clustering (Jedidi, Krider, and Weinberg 1998), sequential products (Lehmann and Weinberg 1998; Prasad, Mahajan, and Bronnenberg 1998), contract design (Swami, Lee, and Weinberg 1998), scheduling (Swami, Eliashberg and Weinberg 1999; Eliashberg, Swami, Weinberg and Wierenga 2001) and the impact of advertising (Zufryden 1996). This thesis advances the above stream of research in marketing of movies, by considering the problem related to retailing of movies.

Typically, perishability is thought of in terms of physical deterioration of a product such as a grocery item with an expiry date. In some sense, this view comes from the supply side of the product. In contrast, in this thesis we adopt a "demand side view" in the context of perishability. Thus, the physical product in the problems considered remains the same, but its demand perishes over time. In recent years considerable work has been done on the treatment of perishability in inventory control (e.g., Abad 1996, Jain and Silver 1994). In inventory control, perishability refers to the physical deterioration of units of a product. For the movies, we analogously define perishability as the decrease in the value/appeal of a movie with the passage of time. As mentioned earlier, previous research in shelf space management and dynamic decision making has mainly focused on non-perishable goods. In the context of movies, the complexity introduced by perishability in dynamic shelf space

management problem can be summarized as: which motion pictures to choose to show each week and for how long to play them. In case of exhibitors with multiple screens, issues such as allocation of different movies to different capacity screens, switching of the movies between screens and multiple screening of same movie add to the complexity of the general problem. This decision-making problem is further complicated by additional factors specific to the movies context. We discuss such factors in detail in the later chapters to follow.

The exhibitors, who are the retailers in movie market, also face the problem of limited "shelf-space" which is the available screens for playing movies. The problem has two parameters: the number of screens available with the exhibitor, either one or multiple, and the nature of perishability of demand of the movies, which can either be stochastic or deterministic. Deterministic dynamic programming, or an integer program, readily models a single screen deterministic demand problem. The incorporation of stochasticity, however, requires more specialized techniques.

In the proposed research we address exhibitor's problem in two ways. We propose an application of "Markov decision process" (MDP) assuming a single screen with the exhibitor and considering stochasticity in the demand of movies. The complexity in this problem comes from the stochasticity in the demand of movies and the perishability of this product in terms of declining demand. To test the proposed approach we compare its performance with some other heuristics. We also investigate the optimal replacement policies under different simulated scenarios and characterize the form of policies that emerge. We explore the possibility that the optimal policy possesses a simple structure in movie replacement problem.

Next we address the problem of exhibitor with multiple screens. We assume an exponentially declining demand pattern of each movie for this multiple-screen problem. The complexity in such problems is because of multiple unequal capacity screens and the decay of movies which causes the exponentially declining demand pattern. With an assumption of equal screen capacities, the problem is similar to the conventional parallel machine scheduling problems. Some work has been done in this regard particularly focusing on the movie exhibitors (Swami, Eliashberg, and Weinberg 1999). However this assumption of equal screen capacities poses serious

limitation in implementation of such models. Considering different capacities for screens introduces nonlinearity into the resulting model and makes it less tractable (Saxena, 2000). We propose the use of heuristic based on Genetic Algorithms methodology (Holland 1975, Michalewicz 1992, Grefenstette 1986). We also investigate the improvement of proposed GA-based heuristic over other heuristics.

The rest of the thesis is organized as follows. Chapter 2 reviews the relevant literature. Chapter 3 explains the model for using MDP in the single-screen-stochastic-demand problem. We test the model for various randomly generated scenarios, by use of design of experiments, regression analysis and ANOVA. For these scenarios we also compare the performance of this model, with generally accepted heuristics. Finally we present the implementation of this model by using actual data at a theatre.

Chapter 4 discusses the SilverScreener model proposed by Swami, Eliashberg, and Weinberg (1999) to address the problem of exhibitors with multiple screens with assumptions of deterministic demand pattern of movies and equal capacity of all screens. Saxena (2000) explains the implementation of this model and proposes a conceptual nonlinear model to address this problem with an added constraint of unequal screen capacities. Saxena (2000) also proposes a heuristic based on demand of individual movies, to address the problem of unequal screen capacities. We explain the proposed GA based heuristic and compare its performance against Saxena's (2000) Screen Allotment heuristic. We test the model for randomly generated scenarios and carry out ANOVA to study the impact of various factors on the improvement achieved by proposed heuristics over Saxena's (2000) heuristic.

In Chapter 5, we discuss the conceptual extensions of the SilverScreener model to address some problems in similar areas and propose the use of GA-based heuristics for problems that cannot be addressed by simple extensions of SilverScreener model. We conclude in Chapter 6 by discussing limitations of the current research.

We now present the summary of two main chapters contained in this thesis.

Chapter 3: Movie Replacement Problem: MDP Approach

The chapter deals with a shelf space management problem, with an objective of maximizing retailer revenue by optimum use of his/her limited display space. The retailer at each decision epoch makes a decision of which product to display. The chapter assumes a single shelf for display with the retailer, which can be allotted to only one product at a time. Each product when displayed will lead to probabilistic demand. The demand for other products will also deteriorate probabilistically with each decision epoch. The stochasticity in the demand for each product induces complexity in the problem of space allocation.

We model this problem as a 'Markov Decision Process' model (Puterman 1994). We define a rank of a product as the expected revenue that it will produce if displayed. We categorize products into groups. Each group has a probability matrix that defines the probability of the product opening into a particular rank and the probability of transition of product from one rank to another, after each stage. For the sake of simplicity we assume these probabilities of transition as static, that is, they do not change with the decision epoch. We also assume independence of these transitions for a product with respect to other product. This means that the transition of a product from one rank to another rank is independent of the transition of other products.

We focus our study on motion picture industry. We investigate optimal replacement policies as suggested by the model under different simulated scenarios for their managerial implications. Next, we characterize the form of the optimal policies that emerge. In particular, we explore the possibility that the optimal policy possesses a simple structure in the movie replacement problem. We also compare the performance of our model with the generally accepted heuristics.

Chapter 4: Multiple Screen Problem: Genetic Algorithm Approach

The analogy of movie screen allocation with the shelf space allocation problem is further explored in this chapter. In practice, retailers allocate space to multiple products rather than a single product. This is also applicable to most movie theaters in

major cities of the world that are multiplexes, that is, theaters with multiple number of screens. The retailers in such situations have to make two decisions. First, which products should be displayed, and second, how much space to be allotted to each product. The complexity in this problem comes from the multiple display spaces available with the retailer. The movie exhibitors have multiple screens with unequal capacities. For the sake of simplicity, we assume a deterministic demand pattern (with random error term) of movies in this case. The number of visitors for a movie scheduled at a particular screen in a week is limited to the minimum of the screen capacity and the general demand for the movie in that week. This situation is analogous to any other retail space allocation problem. With a particular revenue function, which considers the contract term between the movie exhibitor and the distributor the revenue corresponding to movie allocation to each week-screen slot can be calculated. We propose a Genetic Algorithm based heuristic solution to this problem. We compare the improvement in the objective function value (here, total revenue) shown by the proposed heuristic over some other heuristics under different settings of input factors such as the capacity of screens, the contract term between exhibitor and distributor and the number of rapidly decaying movies in a season. We find that the proposed GA based heuristic shows an improvement ranging from 3% to 60% under certain settings of contract terms, type of movies released in the season and the capacities of screens. This analysis suggests some actions for the movie retailers and distributors that we discuss in managerial implications of the proposed research.

CHAPTER 2

LITERATURE REVIEW

Previous researchers have recognized the interface between marketing and operations as an important research domain in the quantitative modeling research stream. In the operations management literature, Acquilano and Chase (1991, p. 17) mention that marketing specialists need an understanding of what the factory can do relative to meeting customer due dates, product customization, and new product innovation. In service industries, marketing and production often take place simultaneously, so a natural mutuality of interest should arise between marketing and OM [operations management]. Karmarkar (1996) stresses the need to do more integrative research between marketing and operations management. This thesis addresses such needs and uses both marketing and operations management tools in solving management problems.

2.1 Shelf Space Management Problems

Shelf (display) space management is a critical issue to the retailers. First, consumers choose from the products that are displayed on the shelf. In this sense, shelf space management determines the ultimate profitability of the retailer. Second, most retailers have limited shelf space to display their products. Therefore, the choice of which products to display becomes an important retailing decision. The decision is particularly complex in those industries in which many new products are continually introduced in market. Finally, anticipating and adapting to dynamic changes in consumer's tastes and demand are key concerns to most retailers. Moreover, such decisions have to be made for several product categories. Though previous researchers (Corstjens and Doyle 1983; Bultez and Naert 1988; Borin, Farris, and Freeland 1994) have addressed some of these issues, they have generally been in the context of packaged goods in a supermarket chain setting. However, other industry

settings, such as the motion picture (or movie) industry considered in this thesis, pose different challenges and intriguing problems.

2.2 Perishability

One of the interesting and important attributes of the motion pictures is their perishability. In recent years considerable work has been done on the treatment of perishability in inventory control (e.g., Abad 1996, Jain and Silver 1994). In inventory control, perishability refers to the physical deterioration of units of a product. For the movies, we analogously define perishability as the decrease in the value/appeal of a movie with the passage of time. As mentioned earlier, previous research in shelf space management and dynamic decision making has mainly focused on non-perishable goods. In the context of movies, the complexity introduced by perishability in dynamic shelf space management problem can be summarized as: *which motion pictures to choose to show each week and for how long to play them.* This decision-making problem is further complicated by some other factors specific to the movie context. We discuss such factors in detail in the later chapters.

2.3 Dynamic Decision Making Problems and Markov Decision Process

The dynamic decision making problems fall under a class of problems known as sequential decision model (Puterman 1994). In these problems, at a specified point in time, a decision-maker, or controller, observes the state of a system. Based on this state, the decision-maker chooses an action. The action choice produces two results: the decision-maker receives an immediate reward, and the system evolves to a new state at a subsequent point in time according to an equation of motion determined by the action choice. At this subsequent point in time, the decision-maker faces a similar problem, but now the system may be in a different state and there may be a different set of actions to choose from. The objective of the decision-maker is to choose a sequence of actions, which causes the system to perform optimally with respect to some predetermined performance criterion.

When the decision is assumed to be made at discrete points in time and the equation of motion is assumed to be deterministic, the problem is usually modeled as a (deterministic) dynamic programming problem (Bellman 1957). The idea behind the dynamic programming approach is to breakdown a large problem into several smaller stages -- called separability -- to devise a sequence of simpler optimal solutions that lead to the overall optimal solution. It is also possible to write an equivalent linear program for a finite state deterministic dynamic programming problem (Bertsekas 1987; Ahuja, Magnanti, and Orlin 1993). This equivalence between dynamic and linear programming allows for the problems involving binary decision variables to be formulated as integer programs.

The Markov decision process is one of the sequential decision models (Puterman 1994). The MDP considers stochasticity in the transition of system from one state to another. Unlike deterministic dynamic programming, the final state of system is not determined exactly by the action taken by the decision maker. With an initial state and action by decision maker, the system has a probability distribution for its evolution to the final states.

2.4 Integer Programming Approach

A linear programming problem in which some or all the variables must take non-negative integer values is referred to as integer linear programming problem. When all the variables are constrained to be integer, it is called a pure integer-programming problem, and in case only some of the variables are restricted to have integer values, the problem is said to be a mixed integer-programming problem. In some situations each variable can take on the values of either zero or one; such problems are referred to as zero-one programming problems.

Integer programming is a valuable tool in operations research having a good potential for applications. Such problems occur quite frequently in business and industry. All the assignment and transportation problems are integer-programming problems. In these types of problems the decision variables are either zero or one.

In all such situations, the decision variable X_j,

$X_j =$ 1 if j^{th} activity is performed,

0 if j^{th} activity is not performed

In addition, all allocation problems involving the allocation of men and machine give rise to integer programming problems, since such commodities can be assigned in integers and not in fractions.

The integer programming problems are solved by either the Cutting Plane or Branch and Bound method. In Cutting Plane method by Gomory (1958), we first solve the integer programming problem as ordinary L.P. problem and then introduce additional constraints one after the other to eliminate certain parts of the solution space until an integral solution is obtained.

In Branch and Bound method, the problem is first solved as a continuous L.P. problem ignoring the integrality condition. If in the optimal solution some variable, say X_j is not an integer, then

$$X_j^* < X_j < X_j^* + 1$$

where X_j^* and $X_j^* + 1$ are consecutive non-negative integers. It follows that any feasible integer value of X_j must satisfy one of the two conditions, namely

$$X_j < X_j^* \text{ or } X_j > X_j^* + 1.$$

These two conditions are mutually exclusive and when applied separately to the continuous L.P. problems, form two different sub-problems. Thus the original problem is "Branched" into two sub-problems. Each of these sub-problems is then solved separately as a linear program, using the same objective function of the original problem. If any sub-problem yields an optimal integer solution, it is not further branched. However if it yields a non-integer solution, it is further branched into two sub-problems. This branching process is continued until each problem terminates with either integer valued optimal solution or there is evidence that it can not yield a better one.

2.5 Parallel Machine Scheduling

A number of machines in parallel are a setting that is important from both the theoretical and practical points of view. From the theoretical viewpoint, it is a generalization of the single machine. From the practical point of view, it is important

because the occurrence of the resources in parallel is common in the real world. One may consider scheduling parallel machines as a two step process. First, one has to determine which jobs are to be allocated to which machines; second one has to determine the sequence of the jobs allocated to each machine (Baker 1993, Pinedo 1995).

In the parallel machine scheduling, the objectives are specified in terms of two performance measures, which are as follows.

2.5.1 Minimizing Makespan

The makespan C_{max}, defined as max (C_1, .., C_n) where C_1,... C_n are the completion time of the jobs, is equivalent to the completion time of the last job to leave the system. A minimum makespan usually implies a high utilization of the machine(s).

McNaughton (1959) presented an elementary result for the makespan problem when the jobs are independent and preemption is permitted. With preemption allowed, the processing of a job may be interrupted and the remaining processing can be completed subsequently, perhaps on a different machine.

The minimum makespan M* is given by

$$M^* = \text{Max}\ \{\frac{1}{m}\sum_{j=1}^{n} t_j,\ \max_j |t_j|\}$$

Where m = number of machines

n = number of jobs

t_j = processing time of job j

2.5.2 Minimizing Mean Flow Time

The other performance measure is minimizing the mean flow time. The flow time is the amount of time job j spends in the system. Flow time is given by,

$F_j = C_j - r_j$ and mean flow time is given by

$$\overline{F} = \frac{1}{n}\sum_{j=1}^{n} F_j$$

where C_j is the completion time and r_j is the release time.

Flow time measures the response of the system to individual demands for service and represents the interval a job waits between its arrival and its departure. In the current study, our objective is neither of the two objectives discussed above, but to maximize the total cumulative revenues generated over the time horizon.

2.6 Decision Support Models in Marketing

Over the last three decades, decision support modeling has flourished in marketing. Today one can name many successfully implemented decision support models in marketing, such as PROMOTIONSCAN (Abraham and Lodish 1993), Rangaswamy, Sinha and Zoltners's (1990) model on sales force restructuring, SHARP (Bultez and Naert 1988), ARTS PLAN (Weinberg and Shachmut 1972), BRANDAID (Little 1972), CALLPLAN (Lodish 1971), and so on[1]. These models have addressed various aspects in marketing at different levels of a supply chain. For example, CALLPLAN efficiently allocates salesforce to customers and products. ARTS PLAN helps manager plan a series of performing arts presentations. PROMOTIONSCAN helps manager in developing and evaluating short-term retail promotions. SHARP model helps retailers decide on shelf-space allocations. The SilverScreener model (Swami, Eliashberg, and Weinberg 1999), presented in Chapter 4, is similar to PROMOTIONSCAN and SHARP models, its aim is to help retailers (of the motion picture industry) improve their decision-making.

Marketing decision support systems (MDSS) (Little 1979, p. 9) are intended to assist decision-makers in taking advantage of available information. Decision-makers should benefit from the availability of more or better data by incorporating the information derived from these data into their decision processes (Blattberg and Hoch, 1990). For this purpose MDSS contain marketing models that make it possible to perform so-called what-if analysis (Wierenga et al. 1994). Blattberg and Hoch (1990) report that a combination of model and manager often outperforms either of these two alone. Hoch (1994) attributes this to the relative strengths of models, which compensates for the relative weaknesses of managers. In general, some of the ways in

[1] Many commercial models based on the similar concepts are now also available in the market.

which the use of decision support models can aid a marketing executive are the following (Montgomery and Weinberg 1973):

- Helps to better utilize a manager's judgment,
- Requires an explicit listing of input assumptions which leads to more informed discussion,
- Provides a method for quick and convenient evaluation of the consequences of alternative plans,
- Allows the emergence of unexpected solutions which open up new areas of problem-solving,
- Expands the range of questions which can be answered by use of the notion of derived judgment,
- Distills from available data relevant information as in new product forecasting,
- Provides a basis for relating marketing inputs to market results and, hence, serves as basis for marketing planning, and
- Diagnoses, based on early data, the adequacy of a market plan and locates areas needing improvement.

Inspite of these benefits, one must recognize the following important points about decision support models. First, models are an aid to the decision-maker, not a replacement. Marketing models often can help the manager make a better decision, but models do not make executive decisions by themselves. Second, models should be proposed as useful tools to the end-user (i.e., manager), not as academic curiosities. Finally, models can be useful to managers in many different ways, and may offer opportunities for efficiency in a broad range of managerial activities.

2.7 Genetic Algorithms

The Genetic Algorithm approach was first proposed by Holland (1975). The Genetic Algorithm seeks to mimic the approach of nature in the evolution of species, that is, it is based on the principles of population genetics to guide the search which results in "survival of the fittest'. This approach requires the specification of the candidate

solutions in a binary or a nonbinary string format. These strings are analogous to chromosomes in nature. A chromosome in turn is composed of genes, each of which can take a number of values called alleles. Thus, for instance, we have the hair color gene at a specific position or locus in the string, which can take a specific allele value black, brown or red. In each generation (iteration) the selection of candidate solutions to participate in the creation of offspring (new candidates) is based on their ability to survive in the competitive environment (that is, its fitness). The GA approach has defined operators to perform activities corresponding to the creation of new chromosomes as a result of mating and chromosome modification as a result of mutation.

GAs have been applied in variety of fields. Goldberg (1989) has used GA for optimization of pipeline systems; Cleveland and Smith (1989) for scheduling flow shop releases; Liepins and Potter (1991) use GA for multiple fault diagnosis; Balakrishnan and Jacob (1996) use GA for product design; Deb, Chakraborty and De (1999) use GA for model based object recognition; Joshi (2000) uses GA in search space of hypercubes.

CHAPTER 3

MOVIE REPLACEMENT PROBLEM: MDP APPAROACH

3.1 Introduction

Movie making is a large, expensive and risky business especially in North America. Hollywood's major studios produce over 250 feature films each year. These films are exhibited on more than 37,000 theater screens in the U.S. and Canada. The average movie costs $40 million to produce and an additional $20 million to market. While some movies do become "blockbusters," more than a half of the movies produced by Hollywood do not recoup their investment even after their release to foreign markets, cable, and television (Vogel 1994). U.S. and Canadian box office revenues have continuously been rising in the past decade to exceed $7.7 billion in 2000; however, costs have been rising faster than revenues. Thus, improvements in operations and decision-making can result in major savings for an exhibitor.

Every week, motion picture exhibitors have to decide whether to replace the movies playing at the screens in their theaters. This is particularly difficult in the two major seasons, Summer and Christmas, when a number of movies are released every week by the studios/distributors. Movies are "perishable" products; the box office appeal of major Hollywood movies lasts only a short time. This perishability, and the complicated contracts between exhibitors and distributors add to the complexity of the resulting dynamic shelf-space management problem. We use the term *decay* to explain the intrinsic weekly decline in the box office attraction of a movie playing at a theater (Krider and Weinberg 1998). We use another term, *aging*, to represent the decline in the box-office attraction of a movie if there is a delay (e.g., by a week) in showing that movie. Aging, therefore, gives rise to an opportunity cost for not playing a movie that has already been released in the market.

The weekly availability of movies and a specialized *contract* between a distributor and exhibitor imply that the decisions an exhibitor takes at one point affect

immediate revenue, future decision opportunities, and thus long-term profits. In signing a contract to play a film in its theaters, the exhibitor commits to playing the film for a minimum obligation period, even if consumer demand is weak. Also, the contract splits the box-office gross revenues in a way that favors the distributors in the first few weeks of exhibition, but shifts to the exhibitor's favor later on.[1]

Movies decay and age in an uncertain fashion. This adds to the uncertainty in demand for movies, and makes the task of their replacement highly challenging. For example, if an exhibitor replaces an existing movie with a new one, he/she may be worse off than before if the new movie decays very rapidly. In addition, the opening strength of the movie may vary. Finally, distributors release their movies at different times during a season; all movies are not available for showing at the same time. However, distributors usually indicate release date of their movie well in advance. Thus, it might be beneficial for an exhibitor to continue playing a movie that is not doing well, to retain space for an anticipated blockbuster movie to be released shortly. In summary, the exhibitor's movie replacement problem we address is: ***which motion picture to show each week, when to replace it, and if replaced, with what film.*** As discussed below, our primary goal in this chapter is to understand the nature of the optimal replacement policies for the perishable products, such as movies, in uncertain and dynamic environments. Accordingly, we focus on the case of a single exhibitor managing a single screen theater and address multiple screen effects in next chapter.[2]

To capture the inter-temporal dependencies of the movie replacement problem, we model it as a Markov Decision Process (MDP). At each decision point, in an MDP, the decision-maker observes the system's state and chooses a course of

[1] For example, the modal contract for one exhibitor across more than 100 movies was for the distributor to receive 60%,50%, 40%, and 35% respectively of the box office receipts in each of the first weeks of a movie's run. However, if the box office surged beyond a certain pre-specified level, then the distributor would receive 90% of all receipts above that level. In any case, the exhibitor retains all concession revenues. For a more detailed explanation of the contract, see Swami, Eliashberg, and Weinberg (1999).

[2] The multiple screen theater case is a simple extension of single-screen theater case as far as movie replacement is concerned. This is because as long as equal screen capacity assumption is made, the decision in a multiple-screen case is a vector of movies. This only increases the dimensionality of the problem without changing the nature of the model proposed. In different-capacity multiple screen case, a modification is required in the definition of revenue generated by a movie on a screen (i.e., minimum of potential revenue and screen capacity). This complicates the problem. This problem with the added constraint is addressed in next chapter.

action. This choice affects future system evolution and reward. An attractive feature of the MDP model is that it provides the decision-maker with an optimal policy or strategy, that is, a prescription for choosing the optimal action in any possible future state. However, this feature is also somewhat limiting because it imposes a limit on the dimension of the problems that MDP can solve.

The objectives of this chapter are the following. First, we provide an MDP model for the exhibitor's decision problem. The MDP approach helps us define the decision maker as a "smart" manager as one who takes into account uncertainty in the environment and adopts a dynamic and long-term view in his/her decisions. Second, we investigate optimal replacement policies under different simulated scenarios and investigate their managerial implications. Next, we characterize the form of the optimal policies that emerge. In particular, we explore the possibility that the optimal policy possesses a simple structure in the movie replacement problem. The optimal policy results are used as a benchmark for comparing the profitability of simple control limit policies and heuristic rules. Finally, we present an example of the real-world implementation of the MDP approach. To best of our knowledge, this is the first attempt to apply dynamic decision-making techniques in movie theatre management.

The chapter is organized as follows. Section 3.2 presents the modeling framework. We present the policies that emerge in various scenarios analyses in Section 3.3. In Section 3.4, we compare the MDP model with simple decision heuristics. Section 3.5 presents an example of the application of the MDP model approach in a real-world setting. We conclude in Section 3.6 with a discussion of limitations and extensions.

3.2 The Markov Decision Process Model

3.2.1 Movie Replacement Problem

Every week, motion picture exhibitors make decisions regarding the replacement of the movie playing at their theater. From a modeling perspective, the key features of the movie replacement problem are decay and aging of movies, uncertainty of demand, the specialized contract and the fact that all movies are not available for

showing at the same time. The movie replacement problem is somewhat analogous to the equipment replacement problem (Derman, 1963) with the current exhibited movie occupying the role of the deteriorating machine.

To model the deterioration of a movie at a screen, we introduce a parameter, *the rank*, of a movie. We define the rank as its relative position with respect to other currently available movies in terms of box-office gross revenue. We assume that in any week, a movie can be in one of the ranks $1,\ldots,z$, where 1 denotes the highest rank and z, the lowest[3]. We further assume that knowledge of the rank of a movie provides complete information about its expected gross revenue. In other words, different movies with the same rank have the same expected gross revenue. Previous researchers have also used the notion of ranks to characterize performance of a movie (De Vany and Walls 1997).

We now describe the timing of events in the exhibitor's decision problem (see Figure 3.1). Each Monday morning, after observing the weekend demand, the exhibitor must decide whether to replace the current movie (say, a) and, if so, with which available film (say, b). In making this decision, the exhibitor must take into account the potential revenues of the current and replacement movies, the contract, and the release dates of replacement and future movies. At the beginning of the season, the release dates of all movies for that season are usually known. Randomness enters the decision problem through the reward streams (attendance) of the movies, which follow a probabilistic pattern. We emphasize that if a movie has been released and not shown by the exhibitor, its appeal may decline.

Figure 3.1: Timing of Events in the Movie Replacement Problem

Start of Week t Weekend demand	Decision Epoch t	Start of Week $(t+1)$
FRIDAY - Movie a playing on the coming Friday	MONDAY - Movie a playing - Choose movie b to play	FRIDAY - Movie b playing

We assume that the exhibitor has sufficient information to determine a movie's rank (an indicator of its expected attendance) for the upcoming decision epoch. This appears to be a reasonable assumption. Swami, Eliashberg and Weinberg (1999) report that week ahead forecasts were correlated at 0.96 with actual box office revenues for the movie theater analyzed in their paper.

In the case of an already released movie, the exhibitor might know its rank based on its attendance pattern at this theater in previous weeks, because an additional print might be playing at another theater or from information about its nationwide performance. In case of a new movie, sources of information for the exhibitor to be confident of a movie's opening include advertising levels, word-of-mouth, private market research, and so forth.

The transition between ranks over time is represented by a Markov process, which is designed to replicate the empirical patterns found in Jedidi, Krider, and Weinberg (1998). Also, based on Jedidi, Krider, and Weinberg's (1998) results, we assume that after the release of the movie, its rank is non-improving, that is, it does not make transition to a higher rank

We now formally state our modeling assumptions:

1) The theater has one screen.
2) The opening rank of a movie is known before its release.
3) Replacement decisions are made on a weekly basis. Further, the replacement decision for the coming weekend is made on the previous Monday, and delivery of the replacement movie occurs instantaneously.

[3] Note that we augment the rank designation when we formulate our model below.

4) The release dates of the movies considered during the planning horizon are deterministic and known in advance.
5) Movies once replaced at a theater are not available for subsequent screening at the same theater.
6) The probability of a transition of a movie between ranks is stationary over time and independent of the other movies.

Assumption 3, 4, and 5 are consistent with industry practice. Assumptions 1 and 6 have been made for the tractability of the analysis. Assumption 6 might be limiting because of the following reasons. The independence between the transition probabilities of movies may be questionable in some cases, because decay of a movie may affect that of another. Second, the transition of the movie between the ranks may vary depending on whether the movie is in the early or later part of its run. We make this assumption to keep the analysis simple at this level of generality. The relaxation of the assumptions will provide attractive future research opportunities.

3.2.2 Model Formulation

A Markov decision process (Puterman, 1994) often referred to as an MDP, is a model for a stochastic sequential decision process. To formulate an MDP we specify decision epochs, states, actions, rewards and transition probabilities. In our application, the exhibitor notes the movie currently playing, its length of play and the ranks of all movies that could be chosen to replace it. The exhibitor then decides whether or not to replace the movie and if so, with which available film. As a consequence of this decision the exhibitor receives a random stream of daily revenues throughout the current week and all movies, which have already opened age and decay in a probabilistic fashion. By solving the MDP, the exhibitor obtains a *policy* or contingency plan which specifies which action to choose each week given any possible configuration of movie being played, its length of play and ranks of available replacements, so as to maximize expected total revenue over the planning horizon.

Before formally defining the model, we introduce the following notation. Recognizing the complexity of the notation, we illustrate and describe it following the definitions.

W -- length of planning horizon, in weeks, say summer season,

M -- set of movies available during the planning horizon, $M=\{m_1, m_2, m_3, \ldots, m_N\}$,

N -- total number of movies considered during the planning horizon, $N=|M|$,

R -- set of ranks of a movie, $R=\{0, 1, \ldots, z, z+1\}$,

R^N -- set of all possible combinations of ranks of the N movies in the set M,

r – a typical vector of ranks of all movies

OPD -- the length of the obligation period,[4]

m -- index number of the movie playing at a decision epoch, $m \in M$,

n -- total number of weeks the movie m has played before a decision epoch,

Z -- vector of ranks, at a decision epoch, of the movies considered during the planning horizon,

$E[NR(s,a)]$ -- expected net revenue to the exhibitor at a decision epoch given state s and action a,

$p_j(i_2|i_1)$ -- probability of transition of movie j, when it is available, from the rank i_1 to i_2, where $i_1, i_2 \in \{1, \ldots, z\}$,

$p_j^{in}(i)$ -- the probability that movie j opens in rank i, $i \in \{1, \ldots, z\}$,

t_{m_1} to t_{m_N} -- release dates of the movies in M during the planning horizon.

We now explain the notation in detail. The length of the planning horizon, W, is finite and measured in weeks.[5] An example of a planning horizon is the summer season, which extends in North America from the last week of May (Memorial Day in the U.S.) to the first week of September (Labor Day in the U.S.). The set of movies, M, consists of a list of the movies available at any time throughout the planning horizon. For example, if three movies, a, b, and c, are available, then the first movie, $m_1 = a$. Similarly, $m_2 = b$ and $m_3 = c$. Consequently, $M = \{a,b,c\}$ and the total

[4] The length of the obligation period is proposed as movie-independent for the case of simulation analyses to be introduced later. However, the model can be readily extended to make this parameter movie-dependent.

[5] We assume that the last decision is made at decision epoch W. Since we do not know future film availability beyond the planning horizon, we choose to formulate a finite horizon problem. The consequence of doing this is that the terminal condition might affect optimal policies in some cases.

number of movies, N, equals three. The set of movie ranks, R, lists the possible ranks an available movie can assume from 1, the best rank to z, the worst rank. For example, in a ranking system *{High, Medium, Low}*, *High* = 1, *Medium* = 2, *Low* = 3, and consequently, $z = 3$. We augment the ranks, 1 to z, by two levels, 0 and $z+1$, to denote the movies that are not available for play. If a movie has not been released, its rank is 0, and if it has been replaced after having been shown by the exhibitor, then its rank is set to $z+1$. The set, R^N, gives all possible combinations of ranks of N movies. For example, if $N = 3$ and $R = \{0,1,2,3\}$, then $R^N = \{(0,0,0); (0,0,1); (0,0,2); (0,0,3); (0,1,0); (0,1,1);\ldots; (1,0,0); (1,0,1);\ldots; (2,0,0); (2,0,1);\ldots; (3,0,1);\ldots;(3,3,3)\}$. The vector of ranks, Z, of various movies has dimension N and denotes the ranks of various movies at a decision epoch.[6] It is clear that Z can assume any value in R^N at any decision epoch. In the above example, if Movies a and b have not been released, and Movie c is available in Rank 3, then $Z = (0,0,3) \in R^N$.

We now provide our MDP formulation of the movie replacement problem. We explain each component of this apparently complex model by an illustrative example in Section 3.2.3. The approach to evaluating a policy for implementation is discussed in Section 3.3.

Decision Epochs (beginning of every week):

$$T = \{1,2,\ldots,w,\ldots,W\}, W < \infty .$$

States (the movie playing, its play length after obligation period, and the ranks of all the movies at a decision epoch):

$$S = \{m,n,Z\},\ m \in M,\ n \in \{-OPD, -OPD+1,\ldots,0,1,2,\ldots, W-OPD\}, Z \in R^N.$$

Actions (continue or replace a movie when possible):

$$A_{(m,n,Z)} = \begin{cases} m & \text{if } n < 0, \text{or } Z_j \in \{0, z+1\} \textit{ for all } j \in M - \{m\} \\ \{j \in M;\ Z_j \notin \{0, z+1\} & \textit{if } n \geq 0 \textit{ and } Z_j \notin \{0, z+1\} \textit{ for all } j \in M - \{m\} \end{cases}$$

However, we did not observe any systematic effects by varying the planning horizon length in this study.

Rewards (expected net revenue to exhibitor in week w):

$$r_w(s,a) = E[NR(s,a)], \quad w = 1,\ldots,W, \quad s \in \boldsymbol{S}, \quad a \in \boldsymbol{A}_s .$$

$$r_{W+1}(s) = 0, \quad s \in \boldsymbol{S}.$$

Transition Probabilities:

$p_w(s_2 \mid s_1, a) =$

$$\begin{cases} \prod_j p_j(Z_{j,2} \mid Z_{j,1}), & \text{if } \begin{array}{c} (a = m_2 = m_1, n_2 = n_1 + 1, Z_{j,2} \geq Z_{j,1} \geq 0, \textit{for all } j \in M) \text{ or} \\ (a = m_2 \neq m_1, n_2 = -OPD, Z_{m_1,2} = z+1, Z_{j,2} \geq Z_{j,1} \geq 0, \textit{for all } j \in M) \end{array} \\ 0, & \text{otherwise,} \end{cases}$$

where

$$p_j(Z_{j,2} = 0 \mid Z_{j,1} = 0) = \begin{cases} 1, & \text{if } w < t_j - 1 \\ 0, & \text{otherwise, and} \end{cases} \qquad (\textit{movie not released}) \quad (3.1)$$

$$p_j(Z_{j,2} = k \mid Z_{j,1} = 0) = p_j^{in}(k), \quad w = t_j - 1, \quad k \in \{1,\ldots,z\}, \quad Z_j \neq 0 \ \forall\, w > t_j - 1,$$

(*movie to be released next week*)

$$p_j(Z_{j,2} = z+1 \mid Z_{j,1} = z+1) = 1 \qquad (\textit{movie already played and replaced})$$

(3.2)

for $\quad s_1, s_2 \in \boldsymbol{S}, s_1 = (m_1, n_1, Z_1), s_2 = (m_2, n_2, Z_2); a \in \boldsymbol{A}_s; w = 1,\ldots, W.$

3.2.3 An Illustrative Example

We begin with the hypothetical movie release scenario shown in Figure 3.2. We assume three movies with two possible ranks for each.

[6] $Z_{j,t}$ denotes the rank of movie j at decision epoch t.

Figure 3.2: Movie Release Pattern for Illustrative Problem

Movie Released	a			c	b			
Week	1	2	3	4	5	6	7	8

Figure 3.2 shows that Movie a is released and available in the first week. Movies c and b become available in Weeks 4 and 5 respectively, that is, t_a=1, t_b=5, and t_c=4. We choose the following values for the other parameters: $W = 8$, $M = \{a,b,c\}$, $N = 3$, $R = \{0,1,2,3\}$, $OPD = 2$,

$$P_a = \begin{pmatrix} 0.3 & 0.7 \\ 0 & 1 \end{pmatrix},\ P_b = \begin{pmatrix} 0.5 & 0.5 \\ 0 & 1 \end{pmatrix},\ P_c = \begin{pmatrix} 0.1 & 0.9 \\ 0 & 1 \end{pmatrix}, \text{ and}$$

$$p_a^{in} = (0.2 \quad 0.8),\ p_b^{in} = (0.7 \quad 0.3),\ p_c^{in} = (0.8 \quad 0.2).$$

In this example, the length of planning horizon is eight weeks. The obligation period is 2 weeks for all movies. The movies can be in the ranks 1 (high) or 2 (low) when they are available. Rank 0 implies that a movie has not been released and Rank 3 implies that it has been replaced after having played. The transition and the initial probabilities are specified for ranks 1 or 2. We express them in matrix format for simplicity. For example, the matrix P_a has components $p_a(j|i)$ which denote the probability that movie a occupies rank i at the next decision epoch given that it occupies rank j at the current decision epoch. For example, $p_a(2|1) = 0.7$ means that there is a probability of 0.7 that movie a declines from rank 1 to rank 2 in any week. These probabilities imply that Movie a is the weakest of the three movies. This is because its initial probability of opening in Rank 1, the better rank, is low (0.2), and even if it does, there is a high probability (0.9) that it decays to Rank 2. The expected net revenue generated by a movie at a decision epoch depends on its rank and the number of weeks it has played before that epoch. The expected net revenue to the exhibitor, presented below, has been generated by assuming common contract terms, which are representative of the industry practice.

Rank \Week	1	2	3	4	5	6	7	8
1	290	290	290	290	290	290	290	290
2	60	70	80	90	90	90	90	90

An explanation of the MDP model follows. Suppose that we observe the system in Week 6 and find that Movie c has been playing for two weeks after replacing Movie a in Week 4. Therefore, $m_6 = \{c\}$, and $n_6 = -2 + 2 = 0$, because the obligation period is two weeks. Further, suppose that the ranks of Movies c and b are 1 and 2 respectively in Week 6, which implies that $r_6 = (3,2,1)$. The rank of Movie a is 3, because it has been replaced. Thus, the state variable, s_6, $= (c,0,(3,2,1))$.

The action specifies whether to play the current movie for an additional week or to replace it and if so, with which other available movie. This decision option depends on which movies are available for replacement, and whether the movie currently playing remains in obligation period. The only action is to continue playing the current movie if either there is no movie available for replacement, or the current movie is in its obligation period. If both of these conditions are false, then the action set consists of both continue and replace decisions. In the example problem, the feasible action in the first three weeks is a, because no other movie is available. In Weeks 4 and 5, since Movie a is replaced (its rank is 3), and Movie c is still in its obligation period (n_4 and $n_5 < 0$), the set of feasible actions is $\{c\}$. From Week 6 onwards, because Movie c has played longer than its obligation period, the set of feasible actions is $\{b,c\}$.

The reward earned by the exhibitor at decision epoch w is the one-period expected net revenue received by the exhibitor by taking action a_w in state s_w.[7]. We assume that the rank of a movie determines its expected gross revenue. We set the reward at period $W+1$ (i.e., salvage value) to zero[8]. Suppose Movie a is playing in

[7] For the sake of simplicity, we do not provide the reward function in this paper, but simply note that it depends on the contract terms, expected gross revenue, and expected concession profits. The exact expression of the reward function is similar to the one used by Swami, Eliashberg and Weinberg (1999). When the box office returns are sufficiently high, the exhibitor retains a constant and not an increasing share of these receipts.

[8] Choice of terminal reward can have a significant effect on the optimal policy. Alternatively we could set the terminal reward equal to the revenue from playing the movie at the last decision epoch indefinitely.

Week 5 (i.e., it has played for a total of four weeks) and the ranks of Movies a, b and c are 1, 1, and 2, respectively. Thus, $s_5 = \{a, 2, (1,1,2)\}$. Then, if either movie a or b is chosen, the one-period immediate expected reward to the exhibitor is \$290; if Movie c is chosen, the reward is \$60.

The probability transition function, $p_w(s_2 | s_1, a)$, specifies the probability that the system will be in the state $s_2=(m_2,n_2,Z_2)$ at the next decision epoch given the current state of the system is $s_1=(m_1,n_1,Z_1)$ and action a is taken. The last component of the state variable, Z_i contains the extended ranks of all movies; as noted before, those that have yet to open have rank zero and those that have already been shown have rank 3. Since the transitions of different movies between various ranks are independent of each other, the transition probability between rank vectors is simply the product of the individual movie transition probabilities. If the action is to continue with the current movie, then the movie index (i.e., the first element of the state variable) remains the same in the next period. The play length of the movie increases by one period. All the movies that have been released either stay in the same rank or make transitions to a lower rank according to their transition probabilities. If instead we choose the action to replace the current movie by another available movie, the first element of the state variable in the next period becomes the movie index of the new movie; the second component, the play length, is set to $-OPD$ denoting that the movie has entered its obligation period. The rank of the current movie m_1 is set to z+1 in the next period. The rest of the available movies make transitions according to their respective transition probabilities. Some movies may not be available at a particular decision epoch because either they have not been released by that week or they have been replaced after having played. The probability expressions for these cases are given by Equations 1 and 2 respectively. Equation 1 implies that if a movie has not been released (i.e., its rank is equal to zero), then its rank remains zero with certainty in the next period until a period before its release. At a period before its release, the movie has an initial probability of opening in the next week in ranks 1 to z. Equation 2 implies that once a movie is replaced (i.e., its rank is equal to z+1), it is never available for screening again. Suppose Movie a is playing in Week 6 and all the movies are in Rank 1. Therefore, $s_1 = \{a,3,(1,1,1)\}$. If we choose action $\{b\}$, then the

probability of system making transition to $s_2 = \{b, -1, (3,1,2)\}$ is $1 \times 0.5 \times 0.9 = 0.45$. Alternatively, suppose Movie a is playing in Week 3 and its rank is 1, then the state is $\{a,0,(1,0,0)\}$. Movie c becomes available in the next week. The probability of transition to $\{a,1,(1,0,1)\}$, therefore, involves the initial probability of Movie c opening in Rank 1 and movie a stating in Rank 1, and equals $0.3 \times 1 \times 0.8 = 0.24$.

3.2.4 Determining Optimal Policies

By a decision rule, we mean a function that specifies the action to use in each state at a specified decision epoch. We use the notation $d_w(s)$ to represent a decision rule. It gives the action chosen in state s at decision epoch w. A policy is a sequence of decision rules, one for each decision epoch. We represent policies by $\pi = (d_1, d_2, \ldots, d_W)$ and compare them on the basis of their expected total reward over the planning horizon. Let $v^{\pi}(s)$ represent the expected total reward if policy π is used and the system is in state s at the first decision epoch. Then,

$$v^{\pi}(s) = E_s^{\pi}\{\sum_{w=1}^{W} r_w(X_w, Y_w) + r_{W+1}(X_{W+1})\} \tag{3.3}$$

where E_s^{π} represents expected value under the policy π when the system starts at decision epoch 1 in state s and X_w and Y_w are random variables, which denote the state and action at decision epoch w, respectively. An optimal policy, π^*, with the largest expected total reward, defines the *value, $v^*(s)$*, of an MDP model through

$$v^*(s) = v^{\pi^*}(s) = \underset{\pi \in \Pi}{\text{Max}} \{v^{\pi}(s), \quad s \in S\} \tag{3.4}$$

where Π denotes the set of all policies. In the next section, we compare and evaluate various alternative scenarios on the basis of the value of their MDP model. We define a *smart* manager as one who uses the optimal policy recommended by the MDP model and achieves the maximal total expect reward. We use the backward induction algorithm (Puterman 1994, pp. 80-82) to determine an optimal policy and value function.

To illustrate these concepts and the benefits of this approach, we interpret the optimal policy for the example problem shown in Table 3.1.

Table 3.1: Optimal Policy for the Illustrative Problem in Section 3.2.3

Decision epoch (w)	State variable (s_w)			Optimal decision rule
	Movie playing	Number of weeks played	Ranks of movies	$d_w^*(s_w)$
1 to 3	a	Corresponding to decision epoch	(1 or 2,0,0)	a
4	a	2	(2,0,1)	c
			Any other combination	a
5 to 8	a	Corresponding to decision epoch	(2,2,1)	c
			(2,1,1 or 2)	b
			Any other combination	a
	b	Corresponding to decision epoch	(3,2,1)	c
			Any other combination	b
	c	Corresponding to decision epoch	(3,1,2)	b
			Any other combination	c

The table shows the state variable, s_w, and optimal decision rule, $d_w^*(s_w)$, to be followed in Week w of the planning horizon.

Weeks 1 to 3 – (Movie a is playing at the beginning of the horizon and no other movies are released until Week 4.) The optimal policy chooses the only feasible action a.

Week 4 – (Movie c becomes available.) Replace Movie a by c only if c opens in Rank 1 and a has decayed to Rank 2 , otherwise play Movie a again.

Week 5 to 8 –

a) If Movie a is playing in Rank 2, then replace it by Movie c if both movies a and b are in Rank 2, otherwise replace it by Movie b if Movie a has decayed to Rank 2 and Movie b is available in Rank 1. In the latter case, the rank of Movie c does not affect the optimal policy. This is because even if Movie c is

available in Rank 1, it is highly likely (with probability 0.9) to decay to Rank 2 by the next week.

b) If Movie b (or c) is playing and beyond its obligation period, then replace it by Movie c (or b) only if the replacement movie is available in a better rank. This implies that if the ranks of the two movies are equal, then it is optimal to play Movie b (or c) again.

3.3 Simulation Analysis

The optimal policy for an exhibitor to follow depends on the quality and quantity of movies available and their release timings. We conduct an experimental study to investigate the nature and magnitude of these effects on optimal policy.

3.3.1 Model Parameters

The input parameters required to generate problems for various simulation analyses are as follows:

(i) length of planning horizon

(ii) number of ranks

(iii) quantity of movies (i.e., total number of movies considered)

(iv) quality of movies

(v) release dates of movies

(vi) expected revenue a movie generates in a given rank

(vii) probabilities of transition of a movie between ranks (including initial probability)

To generate different simulation problems, we assign fixed values to the first two parameters, which are common to all problems. We set the length of the planning horizon to be 8 weeks (e.g., representing the two peak months of Summer season). We consider a three-rank system (e.g., High, Medium, and Low) for the box-office performance of movies. The parameters quantity (iii), quality (iv), and release dates (v) are specific to each simulated problem scenario, and are discussed in a later section. We now discuss a movie industry based study by Jedidi, Krider, and

Weinberg [JKW] (1998), which helps in the operationalization of these and the remaining parameters (vi, vii) of the problem.

JKW modeled the revenue pattern of 102 major motion pictures released in North America during the period December 1990 to April 1992. They used an exponentially decaying function to model the box-office revenue in week t as given by $e^{\alpha+\beta.t+\varepsilon}$, where α denotes a movie's opening strength and $\beta(<0)$ its revenue decay rate. JKW used cluster analysis to segment the movies on the basis of the estimated α and β values, and identified four clusters of films which they referred to as Type 1: Hollywood Heroes (movies that open strongly but decay quickly, e.g., *Awakening, Sleeping with the Enemy*), Type 2: Mega Movies (movies that open reasonably strongly and decay slowly, e.g., *Home Alone, Pretty Woman, Silence of the Lambs*), Type 3: Fast Fades (movies that open weakly and decay very sharply, e.g., *Bad Influence, Lord of the Flies, White Palace*), and Type 4: Fair Flicks (movies that open weakly and decay sharply, e.g., *Joe Versus the Volcano, Internal Affairs, Postcards from the Edge*). The parameters α and β for the four types of movies are shown below.

	Type			
	1	2	3	4
α	-1.484	-1.614	-2.520	-2.255
β	-0.224	-0.110	-0.439	-0.258

Based on the above study, we now discuss the generation of the other parameters of the simulated problems.

3.3.1.1 Exhibitor's Revenue

In the proposed model, the rank of a movie determines expected box-office gross revenue. To operationalize the three ranks, 1) High, 2) Medium and 3) Low, we first assumed a range of weekly box-office revenue from 0 to in excess of 1000 units. To generate the three ranks, we assumed threshold values at 1000 and 500. Thus, a movie which generates more than 1000 in revenue is classified in the high rank, a movie which generates less than 500 units is classified in the low rank, and the remaining

cases are classified as medium. JWK's model is normalized so that the weekly revenues in Ranks 1, 2, and 3 are 1250, 750, and 400 units, respectively.

However, even if two movies are in the same rank, the revenue the exhibitor earns will differ according to the contractual sharing terms, which vary by the number of weeks since a movie's release. To calculate the exhibitor's share of the box-office revenue generated by any movie in a given rank and week (refer to Table 3.2), we chose the sample contract terms used in Swami, Eliashberg, and Weinberg (1999).

Table 3.2: Expected Weekly Net Revenues to the Exhibitor

Rank	Run-length							
	1	2	3	4	5	6	7	8
1	440	440	440	440	440	440	440	440
2	300	360	390	390	390	390	390	390
3	160	180	220	240	240	240	240	240

3.3.1.2 Initial and Transition Probabilities

Initial and transition probabilities are discrete time representations of the exponential declining model. To estimate the values of initial and transition probabilities of each type of movie, we use their respective α and β values and repeatedly generate weekly revenues by varying t from 1 to 50 in the exponential demand model. To induce randomness, we use the error estimates given in Sawhney and Eliashberg (1996)'s study on forecasting of movie revenues.

The initial probability of a particular movie type to open in a particular rank is operationalized by the proportion of time the movies (for a large generation of movies) of this type open in that rank in the first week (refer to Table 3.3 (a)). Similarly the probability of transition of a movie, from rank, say r_1, to another rank, say r_2, is operationalized as the proportion of times the movie decays from rank r_1 from rank r_2 among all transitions from rank r_1 (refer to Table 3.3 (b)).

For example, JKW's clustering results show that Type 1 movies open stronger, but decay quicker than Type II movies. This pattern is preserved in the probabilities generated since Type I movies never opened in Ranks 2 or 3 (corresponding

probability = 0), while 5% of the Type II movies opened in Rank 2 (probability = 0.05). On the other hand, the probability of a Type II movie retaining itself in Rank 1 is higher (0.8) than that of a Type I movie (0.7).

Table 3.3: Probability Matrices for Scenario Analyses

(a) Initial Probabilities

Type of movie	Rank 1	Rank 2	Rank 3
I	1	0	0
II	0.95	0.05	0
III	0	0.56	0.44
IV	0.07	0.93	0

(b) Transition Probabilities

Type I movies

Rank	1	2	3
1	0.7	0.3	0
2	0	0.7	0.3
3	0	0	1

Type II movies

Rank	1	2	3
1	0.8	0.2	0
2	0	0.8	0.2
3	0	0	1

Type III movies

Rank	1	2	3
1	0	0	1
2	0	0	1
3	0	0	1

Type IV movies

Rank	1	2	3
1	0	1	0
2	0	0.6	0.4
3	0	0	1

3.3.2 Simulation Setup

We study the effect of quantity (high/low), quality (high/low), and release timings (high/low early release of quality movies) on the exhibitor's expected revenue using in a complete 3 (treatment) X 2 (level) factorial design.

3.3.2.1 Design of experiments

We examine two levels, high and low, of the *quantity* effect. In the low quantity case, we assume that a single movie is released in every week of the planning horizon (i.e., an 8-movie problem). In the high quantity case, we assume that two movies are released in every week of the planning horizon (i.e., a 16-movie problem).

The *quality* of the movies is operationalized as the number of high quality movies (Type 1 and 2) in a given scenario. To randomize the assignment of a movie to a movie type, we use the relative proportions of Type I (19%), II (7%), III (38%), and IV (36%) movies in Jedidi, Krider, and Weinberg's (1998) sample. Specifically, a random number is drawn among integers between 1 and 100 for each movie. Then the movie is assumed to be Type I if the random number is between 1 and 19, Type II if it's between 20 and 26, Type III if it's between 27 and 64, and Type IV if it is between 65 and 100.

We found in a preliminary analysis that when there are three or more high quality movies (Types 1 and 2) in a scenario, their effect on the value is much more pronounced than the rest of the movies. Therefore, a scenario in which we find three or more high quality movies is classified as *high quality* scenario, otherwise it is classified as a *low quality* scenario.

To operationalize the release date effects, we define an *early release factor* (e.r.f.), which indicates the proportion of high quality movies that are released in first four weeks (i.e., first half of the season).

$$\text{Early Release Factor} = \frac{\text{Number of high quality movies released in Weeks 1 to 4}}{\text{Total number of high quality movies in the planning horizon}}$$

Using a cut-off value of 0.5, we classify a scenario with early release factor of greater than 0.5 as a *high early release effect scenario*, otherwise as a *low early release effect*

scenario. We use a complete 3-way 2-level factorial design for our experiment. That is all 8 combinations of the three factors are evaluated. A summary of the resulting designs is provided later in Table 3.4 (a).

3.3.3 Methodology

Using the above scheme of simulating problems, a total of 66 problems were generated. Based on the input data, the problems were assigned to their respective type of experiment. The MDP model of Section 3.2.2 was coded in C language and run on an Intel Pentium III class computer. The value evaluations were done according to the cumulative expected value criterion presented in Section 3.2.4. Although a small-size problem (e.g. 3 movies, 3 ranks, and 8 weeks) would take only a few seconds to run for the MDP algorithm, the time taken to solve a problem increases drastically with problem size, especially with the number of movies and ranks. In the current analysis, time taken to solve the MDP problems ranged from 3-5 minutes for low quantity cases to 40-45 minutes for the high quantity cases.[9]

Due to the limitation concerning dimensionality in MDP problems, we adopted several state-reduction mechanisms, to speed the execution of the algorithm without impacting its logic. For example, it was observed that the high quality movies (Types I and II) dominate the value of the program. Therefore, we reduced the number of movies in the consideration by including low quality movies (Types III and IV) only if they have not been older than four weeks (i.e., half the season). Similar other mild assumptions were used after checking the robustness of the results.

[9] The time taken to solve a problem depends on the complexity of the problem determined primarily by the number of movies and ranks. For example, in a typical low quantity case (8 movies, 3 ranks, 8 weeks), the number of movie-rank combinations to be considered for each possibility of the state variable are 3^8, or 6561. Since these combination are for each of the eight possible movies playing and their run-lengths (an average of four weeks), the total number of *records* examined in a typical stage of

3.3.4 Simulation Results

Table 3.4 summarizes the simulation results in several ways. Table 3.4(a) gives the mean (and standard deviation) under each factor level combination and Table 3.4(b) gives the mean and standard deviation for each main effect. The ANOVA results in Table 3.4(c) show that the interactions are not significant and that all three main effects (Quantity, Quality and Early Release Factor) are significant at the *.05* level.

Table 3.4 (a): Mean Expected Cumulative Revenues for Experimental Setup

Experiment	Quality[a]	Quantity[b]	Early release factor[c]	Number of problems	Mean expected cumulative revenue*
1	1	0	0	6	1757 (298)
2	1	0	1	6	1939 (419)
3	1	1	0	6	2191 (69)
4	1	1	1	6	2213 (72)
5	0	0	0	6	1247 (247)
6	0	0	1	6	1901 (170)
7	0	1	0	6	1631 (94)
8	0	1	1	6	1854 (422)
				Total = 48**	

a – High (1) setting implies greater than 2 high-quality movies in a season, otherwise low (0) setting

b – High (1) setting implies 16-movie scenario, Low (0) setting implies 8-movie scenario

c – High (1) setting implies an early release factor greater than 0.5, otherwise low (0) setting

* Standard deviation values are shown in parentheses.

** The total number of problems is 48 as 18 out of the randomly generated 66 problems could not be assigned exactly to an experiment type.

backward induction algorithm are of the order of 209952. However, a similar number for a typical high quantity case is of the order of 2754990144.

Table 3.4 (b): Mean Expected Cumulative Revenues for Different Settings of Parameters

Factor	Setting	Mean Expected Cumulative Revenue*
Quantity	High	1972 (164)
	Low	1711 (283)
Quality	High	2025 (214)
	Low	1658 (233)
Early release factor	High	1976 (270)
	Low	1706 (177)

* Standard deviation values are shown in parentheses

Table 3.4 (c): ANOVA Results

Source of Variation	Sum of squares	Degrees of freedom	Mean square	F_0
Quality (A)	1618030	1	1618030	11.1135*
Quantity (B)	876680	1	876680	6.02*
Early release factor (C)	820273	1	820273	5.63*
AB	103253	1	103253	0.709
AC	339495	1	339495	2.33
BC	261528	1	261528	1.79
ABC	55284	1	55284	0.379
Error	2768239	40	58898	
Total	6842782	47		

- Significant at 5 percent.

It can be concluded from the above results that quality of movies released is a major factor affecting the value (expected total revenue) the exhibitor achieves in the planning horizon. Though the effects of quantity and early release factors are also statistically significant, the change in the value is the greatest when the quality changes from a low to high setting. The standard deviation is low for the high setting of quantity and quality factors. This is because in such cases a few movies, usually of high quality, dominate the exhibitor's policy. However, if such movies are available in the early half of the planning horizon (a high early release factor scenario), then the deviation in mean response values would be higher depending on when the high quality movies are released during Week 1 to 4. Note that even though all interactions are insignificant, the Quality-Early Release Factor Interaction has the greatest p-value.[10]

To further investigate the effects of these variables on exhibitor's value, we conducted a simple linear regression analysis on the simulation data. In this analysis, we decompose the effect of quality into two components: (A) mean opening strength of the season (i.e., planning horizon), and (B) mean retention strength of the season. The first variable is a measure of how well the movies opened on the box-office in a particular season. The second variable is a measure of how well the movies were able to retain themselves (i.e., low decay rate) at the box-office. The other two variables considered are number of movies (N, quantity effect), and early release factor (E).

Mean opening strength of the season is operationalized as the mean of the opening strengths (a function of α in the exponential model $\lambda e^{\alpha+\beta . t+\varepsilon}$) of all movies that are released in the season. Similarly, mean retention strength of the season is operationalized as the mean of the decay rates (a function of β in the exponential model $e^{\alpha+\beta\, t+\varepsilon}$) of all movies. Specifically, if N movies are released,

[10] Krider and Weinberg (1998) study the effects of opening strength and decay rate on movie release strategies; consistent with their findings an empirical regression-based analysis of the data in this experiment found that both opening strength and decay rate were significantly related to exhibitor value.

then A and B are quantified as follows.

$$A = \frac{(e^{\alpha_1} + e^{\alpha_2} ......+ e^{\alpha_N})}{N} \quad \text{and} \quad B = \frac{(e^{-\beta_1} + e^{-\beta_2} ....+ e^{-\beta_N})}{N}$$

The resulting regression model with exhibitor's value, V, as the dependent variable, and A, B, N, and E as the independent variables, is given by the following equation.

$$V = b_1 + b_2 * A + b_3 * B + b_4 * N + b_5 * E + \varepsilon \quad ....(3.5)$$

The model results for the simulation data are given below (t-values are in parentheses).

$$\hat{V} = \underset{(-2.7)}{-3099} + \underset{(3.38)}{9376.81} * A + \underset{(2.26)}{4257.09} * B + \underset{(6.73)}{50.42} * N + \underset{(2.14)}{191.77} * E$$

$$R^2 = 0.87$$

where $\hat{V}$ is the mean exhibitor's value for the season.

The model provides a good fit to the simulation data ($R^2 = 0.87$). The significant values of the regression coefficients corroborate the results achieved earlier. In absence of a composite measure of quality, the effect of the quantity is the strongest. However, the results on the two measures of quality further information regarding opening and retention strengths. The results indicate that in the simulated problems, the effect of the opening strength is slightly stronger than the retention strength of the movies in a season. This could be due to the relatively smaller length of the planning horizon, which may also explain the lower (although significant) effect of the early release factor.

3.3.5 Discussion

Intuition suggests that the exhibitor should be better off in a high quantity than a low quantity scenario. The reason is that a high quantity scenario offers the exhibitor increased flexibility in scheduling movies. However, our results show that quality of the movies is more important than quantity. Moreover, the exhibitor should prefer the

high quality movies that are released early in the horizon so that they could schedule them more profitably during the planning horizon. This in turn suggests that the distributors of the strong movies should aim at releasing their movies earlier in the season so that the exhibitor has an incentive to play them longer.

3.4 Comparison with Heuristics

In this section, we compare the MDP approach with two benchmark heuristics. The first heuristic is motivated by theoretical results on structured policies (Puterman 1994, p. 103) and the second heuristic is based on the relationship dilemmas in distribution channels (Swami, Eliashberg, and Weinberg 1999).

3.4.1 Rank-based Optimal Policies and Greedy Heuristics

Establishing the existence of optimal policies with a special structure is one of the principal uses of MDP methods. Such structured policies are useful because they are easy to implement, facilitate computation and allow investigation of the effect of model parameters. Examples of policies with simple structure include (s,S) policies in inventory control models and control limit or critical number policies in queuing control or equipment replacement models. Because of its similarity with the machine replacement problem, we expect that that an optimal policy for the movie replacement problem would have the form: *In a given week, continue currently playing a movie if either it is in its obligation period or its rank is below*[11] *a critical number, otherwise replace it by a new movie.* While we do not establish the optimality of this policy theoretically, numerically derived optimal policies suggest that this might be close to optimal.

We investigate the performance of this policy in the following representative problem of eight weeks, five ranks and two movies (of the same type). The first movie, a (Type I movie), is playing at the beginning of the horizon, and the other one, b (Type II movie), is released at Week 3. The optimal movie replacement policy is presented graphically in Table 3.5.

[11] Recall that a higher rank number of a movie indicates a worse position.

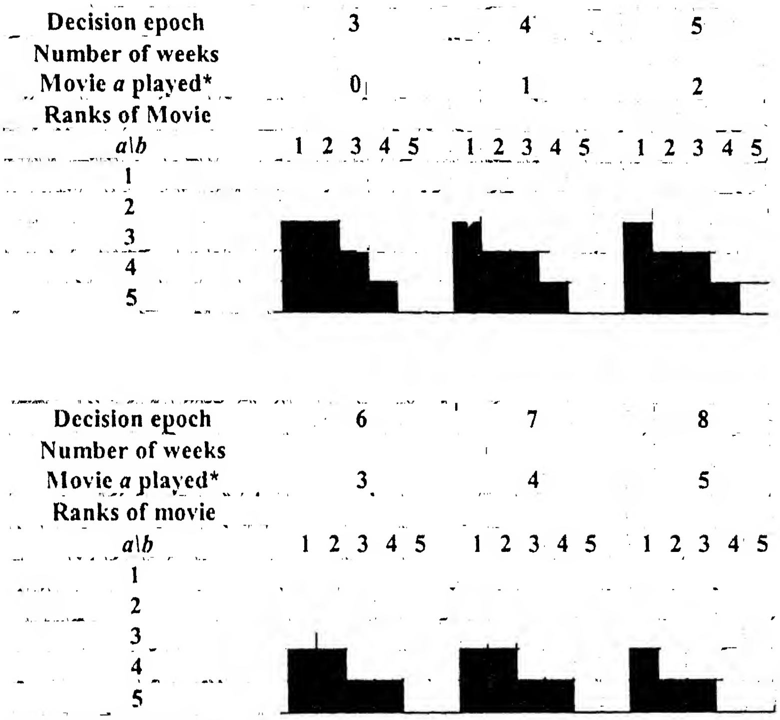
Table 3.5: Results of Rank-Based Optimal Policy Analysis

Movie currently playing = *a* (Type 1)
Movie available for replacement = *b* (Type 2)

Actions

Retain *a*
Replace *a*

Optimal Actions Space



* - Beyond its obligation period of two weeks.

The figure shows that in decision epochs 3-5 if Movie a is in Rank 1 or 2, then it is optimal to continue playing it no matter which rank Movie b opens in and in decision epochs 6-8 it is optimal to continue playing movie a in ranks 1-3. At decision epoch 3 the optimal policy replaces a if its rank is lower than that of b. At subsequent decision epochs, the optimal policy departs slightly from this form.

Similar results for similar other test problems motivate using the following variant of a critical rank heuristic:

Rank Based Heuristic: *Each week, if a new movie becomes available, replace the existing movie only if it is not in its obligation period and either its rank is above a critical value or the rank of the new movie is better than that of the one being currently playing.*[12]

Swami, Eliashberg, and Weinberg (1999) suggest that exhibitors sometimes face a channel relationship dilemma. Distributors want to find screen space for their new movies and pressure the exhibitor to book the latest release. Such an exhibitor uses a decision rule such as:

Distributor Pressure Heuristic: *Each week, if a new movie becomes available, to maintain a good relationship with the distributor, I will replace my existing movie, if it is not in its obligation period, with the new movie. If more than one new movie become available in a week, then I will replace the current movie by the highest ranked movie.* Notice that the manager following this heuristics tries to accommodate the new movies released by all the distributors as long as he/she has "space" to show them. However, with some element of smartness in the decisions, the exhibitor replaces only by the best movies (in terms of ranks) available at any decision point.

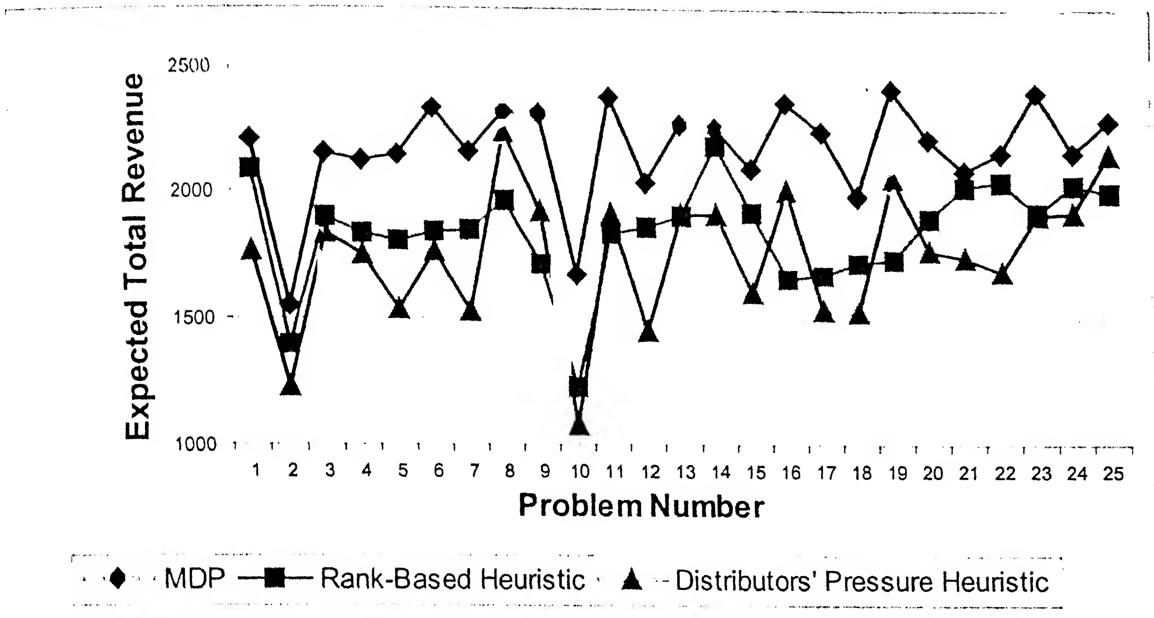
3.4.2 Analysis Results

The comparison of MDP approach with rank based and distributor pressure heuristics was performed on twenty-five 16-movie problems. The expected values from using the heuristics was calculated using backwards induction with a fixed decision rule specified by the heuristic. Figure 3.3 shows the cumulative expected values of the MDP model and the two heuristics.

---

[12] Swami (1998) discusses a version of this heuristic and compares its performance to other decision-making approaches including a benchmark optimal policy.

Figure 3.3: Comparison of MDP Optimal Policy with Heuristic Policies



Expected total revenues for three different policies over 25 problems. Mean for MDP optimal policy is 2170, mean for Rank-Based Heuristic is 1842 and mean for Distributors' Pressure Heuristic is 1751. Observe that the MDP optimal policy dominates the other two in all problem cases.

As can be expected, the mean value of the rank-based heuristic (1842) is higher than that of the distributors' pressure heuristic (1751). This is because in the rank-based heuristic, the exhibitor does not "accommodate" the distributors. This loss in value can be characterized as a "cost" of relationship management. However, the rank-based heuristic performs considerably worse than the optimal policy (2170) for the moderate size problems considered in this analysis. These results points towards the significant advantage that can be gained by using the MDP approach in larger, ongoing decision-making situations because the MDP approach captures more of the subtleties of a dynamic decision situation than simple heuristics.

3.5 Implementation of the MDP Approach

In this section, we discuss how the MDP approach may be applied in a realistic setting. To illustrate this application, we use the data of Swami, Eliashberg, and Weinberg (1999), which was publicly available in *Variety* (1989). The theater considered is the 84th St. Sixplex, a six-screen theater in New York.

The movie scheduling recommendations are developed for the 1989 eight-week summer season. The data available are ex post box-office revenue data, which had to be recast for the MDP approach in the current application. For this purpose, we considered 27 weeks of the year 1989 and a corresponding period from its preceding year 1988. We classified the movies of the two years separately in movie types following Jedidi, Krider, and Weinberg (1998). First, data were median split on the basis of the opening strength of movies. Then, each of these groups were median split on the basis of their decay rates. Thus, we obtained four "types" of movies on the basis of their opening and decay rates for both the years 1988 and 1989. Their corresponding expected revenues, initial and transition probabilities were generated in a similar manner as that followed in the simulation discussed in Section 3.3.1.

To summarize, the transition matrices for the four clusters were estimated using 1988 data. Then, prior to the 1989 season, it is assumed that the exhibitor would know the list of the available movies, their release dates, the cluster that each movie fits in, and the corresponding movie parameters (e.g., probability matrices). However, the exhibitor does not know a priori which rank each movie would actually achieve in a given week.[13]

Applying the MDP model to this problem, we found that in six out of the eight weeks, the model selected the movie that had the highest revenue for that week as recommended by the integer linear programming based model *SilverScreener* of Swami, Eliashberg, and Weinberg (1999); in the other two weeks the MDP model selected the movie that had the second highest revenue. These results, though only illustrative and not conclusive, are encouraging because Swami, Eliashberg and Weinberg's (1999) results partially use ex post box-office revenue data. The current

[13] This approach is based on Swami, Eliashberg and Weinberg (1999) and is also followed in practice by several exhibitors.

results indicate that the MDP approach could mimic the "best" decisions in movie scheduling, where best is defined in terms of optimal movie scheduling done using ex post data. Also, we presented the application for a single eight-week time window. Extensions of the approach for multiple screens and in a "adaptive scheduling" manner (Swami, Eliashberg and Weinberg 1999) could provide solutions in more realistic settings.

3.6 Conclusion, Limitations and Future Research

This chapter addresses the need to do more integrative research between marketing and operations research and applies tools from both fields to solve an interesting management problem. We propose a modeling approach to analyze a retail management problem regarding the replacement of motion pictures on theater screens. The problem is complex because it involves stochastic elements in a dynamic decision making setting. One of the strengths of the MDP formulation is that it helps represent this complex problem by breaking it into separate components, which are easier to model. The resulting model provides theoretical and conceptual insights into the replacement phenomenon of perishable products such as movies. It then addresses some typical decision situations faced by movie exhibitors and provides optimal strategies for them to follow.

The *quality, quantity and availability* analysis suggests that when there are more high quality films, the exhibitor's screen (shelf-space) becomes more valuable. On the other hand, availability of additional low quality movies is not as valuable.

We also investigated the possibility of characterizing the structure of *rank-based optimal policies* as a control limit policy. Our policy results depend on the state of both the current and replacement movies. This creates a two (or multi-)-dimensional decision space whose area depends on the relative attractiveness of the current and replacement movies, and the stage of the planning horizon. These results suggest an easily implementable heuristic. But our evaluation of it shows that the optimal policy from the MDP yields significantly higher revenue.

We now discuss some limitations of our study. We have assumed that rank transition probabilities are known and stationary. It would be interesting in future

research to examine how learning and dynamic updating of probabilities would affect optimal policies.

Another issue concerns the end-of-horizon effects. The end-of-horizon effects are usually summarized by an appropriate salvage value. Though we use zero salvage value in our example problems, the methodology allows any reasonable terminal value. For example we might replace zero by the expected total revenue of the movie playing in the last period.

Finally, we restricted attention to a single screen case. However, interesting issues arise in case of a multiple-screen theater. Some of these issues are allocation of different movies on different capacity screens, switching of the movies between screens, multiple screenings of the same movie, and so on. These are interesting research questions, which we have partially addressed in the next chapter. In addition, other industries pose similar interesting problems involving the allocation of selling space to perishable products with limited lifetimes.

CHAPTER 4

MULTIPLE SCREEN PROBLEM: GENETIC ALGORITHM APPROACH

4.1 Introduction

The previous chapter dealt with a single screen problem. In this chapter we consider a different capacity multiple-screen problem. Most of the movie theatres in the major cities in the world are multiplexes, that is, they have more than one screening rooms. The exhibitor is interested not only in the optimal use of any one of his/her screens but has to optimize the use of all the screens. We propose to address these issues in this chapter. We present use of Genetic Algorithm based heuristic in solving the multiple-screen problem, for which simple mathematical programming approaches do not provide easily tractable exact solutions. Some interesting issues arise in case of a multiple-screen theater. Since the capacity for each screen is different, the decision of whether or not to schedule a movie, is accompanied by a decision of which movie should be allocated to which screen. If only demand of a movie is considered, the allotment may not be optimal because of some other factors like contract term between the exhibitor and the distributor. Again, with the varying demand for each movie and a varying percentage of total revenue that the exhibitor will retain, the movies may have to be switched between screens. The issues like multiple screening of movies (i.e., the same movie displayed on more than one screen in a given week) are other interesting issues arising in this case.

The chapter is organized as follows. In the next section we discuss the previous work specific to the problem. In Section 4.3, we explain in detail the exact problem and briefly explain the SilverScreener model developed by Swami, Eliashberg, and Weinberg (1999) which deals with a similar problem. We also explain the limitation of this model with the multiple-screen formulation proposed by Saxena (2000). We explain by an illustrative example, the situations in which the heuristic proposed by Saxena (2000) might give inferior solutions and justify the use of some

other heuristic. Towards this end, in Section 4.5, we explain the proposed Genetic Algorithm (GA) based heuristic. In Section 4.6, we discuss the computational study and the results obtained. In Section 4.7, we discuss the managerial implications and conclusions of our study. We also discuss the directions for future research.

4.2 Literature Review

In this section we discuss the literature that is very specific to this problem. Swami, Eliashberg and Weinberg (1999) deal with the problem of equal capacity multiple-screen theater scheduling. They propose an integer programming approach to this problem. Saxena (2000) extends their work by implementing the SilverScreener at a theater chain in The Netherlands. Saxena (2000) also addresses the limitation of model, which assumes equal capacity of all screens and proposes a conceptual non-linear model to relax the assumption. To address the issue of unequal screen capacities, Saxena (2000) proposes a screen allotment heuristic. We propose a GA based heuristic for this problem and compare the results of our heuristic with Saxena's (2000) screen allotment heuristic.

4.3 Problem Formulation

4.3.1 The Exhibitor Problem

In this section, we present the space-allocation problem faced by exhibitors, as proposed by Swami, Eliashberg, and Weinberg (1999). The above researchers presented the exhibitor's problem and its formulation in the North American context. Swami, Eliashberg, and Weinberg (1999) present the exhibitor's problem as follows.

Every week, motion picture exhibitors have to make an important decision regarding the replacement of the movies playing at the screens in their theatres. The dynamic environment thus induced gives rise to the notions of decay and aging of movies. Decay is the intrinsic weekly decline in the box-office attraction and gross revenues (grosses in industry jargon) of a movie playing at a theater (Krider and Weinberg 1998). Aging is the decline in the value, that is, gross generating power, of a movie from an exhibitor's perspective if there is a delay (by week) in exhibiting the movie at

the theater. Aging, therefore, results in an opportunity cost of not being able to play a particular movie.

The above scenario is further complicated by the nature of the contract between the distributor and the exhibitor. The basic structure of the contract is fairly standard between different distributor-exhibitor pairs although the individual terms may vary depending on the relationship between the two parties. A typical exhibition contract states a fixed obligation period and a differential revenue sharing scheme in different weeks between the distributor and the exhibitor. The obligation period limits the ability of an exhibitor to replace a movie with less than satisfactory box-office performance in the initial weeks after its release.

In a given week, the revenue sharing scheme splits the gross of a movie between a distributor and exhibitor by one of the two rules: a) 90%/10% over house nut[1] with minimum gross percentage, or b) a movie type based contract term where the percentage revenue share for distributors and exhibitors depend on the type of movie (Swami, Lee and Weinberg, 2000). If the *90%/10% over house nut* rule operates, then the distributor receives 90% of the gross after the exhibitor has deducted and retained the house nut amount. Accordingly, under this rule the exhibitor keeps 10% of the gross over house nut plus the house nut amount. The exhibition contract also contains minimum percentage figures as specified by the distributor for every week of the expected play length of a movie. These figures will be used if the *minimum gross percentage* sharing rule is invoked for revenue sharing. Under this rule, the whole gross box-office revenue amount (without house nut deduction) is split according to the specified percentage for that week. The splitting terms (in favor of distributor and exhibitor, respectively) specified by the distributor under a typical contract may appear as follows (see, for example, Squire (1992), p. 315, for the movie *Batman*).

90%/10% over approved house allowance with minimums of:

First week at 70%/30%

Next two weeks at 60%/40%

[1] House nut is a small, negotiated amount, which the exhibitor receives from the distributor. It does not necessarily bear any relationship to the theater's actual expenses, and is only meant to allow for some cushion in the exhibitor's profit margins.

Next week at	50%/50%
Next week at	40%/60%
Balance at	35%/65%

The manner in which the box-office grosses are split favors the studio (distributors) in the first few weeks of the movie playing, but shifts to the exhibitor's favor later on. Distributors thus have a strong incentive to promote the movies intensively in their initial play period. On the other hand, the longer the exhibitor plays the movie, the larger becomes his/her share of the box-office receipts. At the same time, theater attendance for a movie typically declines the longer the movie plays.

In addition to the revenue earned from the box-office gross, the exhibitor also earns some income from concession sales such as popcorn, candies, and soft drinks. The concession sales, however, depend on individual demands generated by the movies playing at the theater. The exhibitor does not share the concession income with the distributor. Most theaters have multiple movies playing at their multiple screens. Given the complexity of the revenue sharing scheme and the dynamics of movie availability and decision making, the managers of such multiplexes are faced with non-trivial decisions of selecting and scheduling movies on different screens in a fixed planning horizon. In the next sub-section, we discuss a mathematical model SilverScreener proposed by Swami, Eliashberg, and Weinberg (1999).

4.3.2 SilverScreener: The Screens Management Model

4.3.2.1 Assumptions

Swami, Eliashberg, and Weinberg (1999), in order to simplify the exposition, make the following assumptions.

1. The availability of the movies to be released during the planning horizon is known in advance,
2. The weekly revenues to be generated by the movies considered during the planning horizon can be estimated in advance,
3. The replacement decisions are made on a weekly basis,
4. All the screens in the multiplex are of equal capacity,

5. There is no time lag between placing an order for a new movie and its arrival,

Assumptions 1, 3 and 5 are not limiting and, in fact reflect current industry practice. Assumption 2 is a strong assumption about *a priori* knowledge of movie revenues. However, data collected from *Variety* suggest that managers have a reasonable estimate of box-office gross revenue of a movie. Equal capacity screens are assumed (Assumption 4), which addresses the questions regarding which movies to pick and how long to show them. In the model implementation section, we discuss Saxena's (2000) approach to this assumption. We also explain the serious limitation in Saxena's (2000) approach and propose a GA-based heuristic to overcome it.

4.3.2.2 Definition of Variables

W - length of planning horizon,

H - number of screens in the multiplex,

N - total number of movies considered during the planning horizon,

r_j - Release date of movie j,

d_j - due date (if applicable) of movie j,

x_{jiw} - binary (decision) 0-1 variable which takes value 1 if movie j is scheduled for i weeks beyond its obligation period starting in week w,

P_{jiw} - profit received by the exhibitor if x_{jiw} is equal to 1,

$GROSS_{jw}$ - box-office gross revenue generated by movie j in week w,

POP_{jw} - concession profit generated by movie j in week w,

VC_{jw} - variable cost due to movie j in week w,

FC_w - fixed cost of multiplex in week w,

$EXSHARE_{jw}$ - exhibitor's share of box-office revenue for movie j in week w,

OPD_j - obligation period of movie j,

C - house nut.

4.3.2.3 The Model

Time-Indexed Formulation

We now discuss *time-indexed formulation* (Sousa and Wolsey 1992; Williams 1997) that is particularly useful for solving the exhibitor problem. This formulation is based on the idea of dividing the planning horizon $[0,\ldots,W]$ into W discrete intervals of unit length. To model the screen management problem, a binary variable x_{jiw} is defined, which equals 1 if movie j is shown for i weeks *beyond its obligation period* starting in week w of the planning horizon, and 0 otherwise. Notice that the obligation period constraint is included in the definition of x_{jiw} itself. For example, if obligation period of Movie Number 3 is 2 weeks, then $x_{301} = 1$ implies that it is shown for 2 weeks starting in week 1.

The time-indexed formulation highlights some key differences between the exhibitor problem and typical machine scheduling problems. First, all movies do not have to be played in the exhibitor problem, while all jobs have to be scheduled in the machine scheduling problems. Accordingly, the proposed formulation is aimed at helping the exhibitor decide about two critical decision variables: *choice* of which movies and deciding *play lengths* of the chosen movies. Second, the following types of decision-making goals are found to be prevalent in machine scheduling problems: turnaround, timeliness, and throughput. In contrast, the exhibitor problem offers a situation in which the scheduling decisions directly affect profitability, a more relevant decision making goal, both by affecting gross revenue and concession profits. Finally, the main parameters of interest in machine scheduling are the lengths of the jobs, and their release and due dates. The exhibitor problem, on the other hand, deals with additional variables such as complicated contract terms and an exponentially decaying demand function.

Profit Function

P_{jiw}, the total profit the exhibitor receives corresponding to each x_{jiw}, is defined as:

$$P_{jiw} = \sum_{u=w}^{w+SCR_{ji}-1} (-FC_u + POP_{ju} - VC_{ju}) + I_{\theta_{ju}} * \{0.1 * (GROSS_{ju} - C) + C\} +$$

$$(1 - I_{\theta_{ju}}) * EXSHARE_{ju} * GROSS_{ju}, \tag{4.3.1}$$

$$j = 1, \cdots, N, \quad i = 0, \cdots, k_j, \quad w = r_j, \cdots, d_j - SCR_{ji} + 1.$$

where θ_{jw} is a logical condition given by

$$\theta_{jw} = (0.9 * (GROSS_{jw} - C) > (1 - EXSHARE_{jw}) * GROSS_{jw}),$$

and

$$\begin{aligned} I_X &= 1 \text{ if } X = TRUE \\ &= 0, \text{otherwise,} \end{aligned}$$

$k_j = d_j - r_j - OPD_j + 1$ = maximum possible number of weeks movie j can be shown *beyond its obligation period* starting in r_j or any *feasible* week thereafter,

$SCR_{ji} = OPD_j + i$ = total screening period for movie j if it is shown for i weeks *beyond its obligation period*, where $i = 0, \ldots, k_j$

The profit expression proposed by Swami, Eliashberg, and Weinberg (1999) consists of two portions corresponding to the two different conditions of the contract terms[2]. The two conditions, operationalized by the logical variable I, follow directly from the revenue sharing terms described earlier. In addition, there are variables for fixed and variable costs. Fixed costs (e.g., rent, lights, and weekly maintenance) are the costs that the exhibitor has to incur irrespective of the traffic generated by the movies playing in a particular week. Variable costs, such as salaries of the temporary staff hired for a particular movie, vary by movies as well as weeks.

The above operationalization of P_{jiw}'s simplifies the solution procedure considerably since they can now be computed independently of the optimization routine. Further, the variables k_j and SCR_{ji} help us "cover" all feasible (P,x) pairs

for a movie. For example, suppose movie j has parameters: $OPD_j = 2$, $r_j = 1$, and $d_j = 4$. If the movie is scheduled in Week 1 (i.e., $w = 1$), then it can be shown for 2, 3 or 4 weeks, that is, for 0, 1 or 2 weeks beyond the obligation period. Since $k_j = 2$, i varies from 0 to 2. The corresponding values of $SCR_{ji} (= 2 + i)$ are 2, 3, and 4, respectively. Therefore, the corresponding (P,x) pairs are (P_{j01}, x_{j01}), (P_{j11}, x_{j11}), and (P_{j21}, x_{j21}), respectively.

Demand Function

Swami, Eliashberg and Weinberg (1999), Krider and Weinberg (1998), and Sawhney and Eliashberg (1996) use an exponentially declining demand curve to estimate the number of visitors attracted by a movie.

$$\text{Demand (number of visitors)} = VISITOR_{ju} = \alpha_j e^{\beta_j + \varepsilon} \tag{4.3.2}$$

where $\alpha_j > 0$ and $\beta_j < 0$ are opening and decay factors for Movie j, and $\varepsilon \sim$ normal $(0, \sigma^2)$.

Problem Statement

The time-indexed formulation of the exhibitor problem is presented as follows:

$$\max \sum_{j=1}^{N} \sum_{i=0}^{k_j} \sum_{w=r_j}^{d_j - SCR_{ji} + 1} P_{jiw} \cdot x_{jiw} \tag{4.3.3}$$

subject to

$$\sum_{i=0}^{k_j} \sum_{w=r_j}^{d_j - SCR_{ji} + 1} x_{jiw} \le 1, \quad j = 1, \ldots, N \tag{4.3.4}$$

$$\sum_{j=1}^{N} \sum_{i=0}^{k_j} \sum_{q_j = w - SCR_{ji} + 1}^{w} x_{jiq_j} \le H, \quad w = 1, \ldots, W \tag{4.3.5}$$

$$r_j \le q_j \le d_j - SCR_{ji} + 1, \quad j = 1, \ldots, N; \; i = 0, \ldots, k_j \tag{4.3.6}$$

---

[2] The SilverScreener does not consider the movie based contract term, however while comparing the performance of proposed GA-based heuristic with Saxena's (2000) heuristic, which uses SilverScreener in its first step, we made suitable modifications in the SilverScreener model to consider

$$x_{jiw} \in \{0,1\} \qquad (4.3.7)$$

In the above model, Statement (4.3.3) denotes the objective function, which is to maximize cumulative profit over the season. Constraint (4.3.4) ensures that a movie is played in only consecutive weeks. It also allows a movie not to be scheduled at all. The next constraint restricts the total number of movies, scheduled in any week of the planning horizon, to the total number of screens in the multiplex. In doing so, it sums up all the movies, which are released earlier than or in the week under consideration. The set of inequalities denoted by Equation (4.3.6) is an indexing constraint, which restricts the variable q_t in Constraint (4.3.5) to feasible values. Constraint (4.3.7) defines x_{jiw} to be a binary variable.

4.3.2.4 SilverScreener Implementation at Pathe, The Netherlands

Screen Capacity

In their formulation Swami, Eliashberg and Weinberg (1999) assumed all the screens of equal capacity. However in the implementation of the model, Saxena (2000) followed the following simple heuristic for allocating the movies on different capacity screens. While evaluating the performance of GA based heuristic, we will compare its performance with Saxena's (2000) heuristic.

Screen Allotment Heuristic (Saxena 2000)

The statement of the heuristic is:

In a particular week, allocate the movie with highest number of visitors to the highest capacity screen, the movie with the next highest number of visitors to the next highest capacity screen, and so on.

In other words, the application of the model first chooses a set of movies, and then the movies are allocated to the theater screens in the order of their capacities. The screen allocation heuristic is similar to the managerial decision making in some instances.

the movie based contract terms.

4.3.2.5 Illustrative Example to Show Limitation of the Heuristic by Saxena (2000)

We illustrate the limitation of the screen allotment heuristic proposed by Saxena (2000) with a simple example. Consider a two week two screen problem. Suppose that there are three movies available for playing. The demand for these movies for two weeks is as shown below.

Week\Movie	Demand for Movie 1	2	3
1	1700	1200	500
2	200	1000	300

The exhibitor share in net revenue for each movie in each week is assumed as shown below.

Runlength\Movie	Percentage Exhibitor Share 1	2	3
1	10	15	30
2	30	20	50

Let the screen capacities of the two screens be 700 and 200, respectively.

The SilverScreener will select Movies 1 and 2 for the first week and Movies 2 and 3 for the second week. The Screen allotment heuristic proposed by Saxena (2000) will allocate Screen 1 to Movie 1 in first week and to Movie 2 in second week. Similarly Screen 2 will be allocated to Movie 2 in Week 1 and Movie 3 in Week 2. The revenue earned will be as shown below.

Slot	Movie	Revenue
Week 1 – Screen 1	1	700*0.1=70
Week 1 – Screen 2	2	200*0.15=30
Week 2 – Screen 1	2	700*0.2=140
Week 2 – Screen 2	3	200*0.3=60

Thus the total revenue for exhibitor will be 300. A better schedule can be generated as follows

Slot	Movie	Revenue
Week 1 – Screen 1	3	500*0.3=150
Week 1 – Screen 2	2	200*0.15=30
Week 2 – Screen 1	2	700*0.2=140
Week 2 – Screen 2	3	200*0.5=100

This schedule will generate the exhibitor revenue of 420. Thus, the example illustrates the limitation of screen allotment heuristic as proposed by Saxena (2000). Since GA-based heuristics are search procedures we might reach the better solution by employing such techniques.

4.4 Multiple Screen Formulation (Saxena 2000)

SilverScreener model assumes all the screens of equal capacity (Assumption 4) for simplicity. The screen capacity is, of course, a critical variable to the problem formulation only if the demand of a movie exceeds screen capacity. Indeed, the primary reason for the exhibitor to have different screen sizes is that the attendance varies for different movies and having different sizes increases scheduling flexibility (while saving capital cost in the original construction of the theater). A movie may be switched from a higher to a lower capacity screen[3]. We now discuss the extension to the model as proposed by Saxena (2000) for an unequal capacity case.

Definition of Variables

T - Length of planning horizon,

H - Number of screens in the multiplex,

N - Number of movies considered during the planning horizon,

r_j - Release date of movie j

d_j - Due date of movie j

cap_s - Capacity of screen s

x_{jsu} - Binary (decision) 0-1 variable which takes value 1 if movie j in scheduled on screen s in week u

y_{jw} - Binary (decision) 0-1 variable which takes value 1 if movie j in scheduled in week w

P_{jsu} - Profit received by the exhibitor if x_{jsu} is equal to 1

visitor[j,u] - number of visitors for movie j in week u

floor[j,u] - distributor's share in the box-office revenues for movie j in week u

Problem Formulation

Saxena (2000) defines P_{jsu}, the exhibitor receives corresponding to x_{jsu}, as

P_{jsu} = [Min { cap_s, visitor [j, u]}] * (2 + 13.5*0.89725*(1-floor [j, u]))

Since the implementation of SilverScreener by Saxena (2000) was in context of The Netherlands, the profit function is according to the practices of The Netherlands' motion picture industry. The above problem is formulated as an integer-programming problem. The time-indexed formulation is presented below.

Objective function: $$\text{Max} \sum_{j=1}^{N}\sum_{s=1}^{H}\sum_{u=1}^{T} P_{jsu}x_{jsu} \qquad (4.4.1)$$

Subject to:

$$\sum_{s=1}^{H} x_{jsu} \leq 1 \qquad \forall \quad j, u \qquad (4.4.2)$$

---

[3] The opposite case may also occur when a movie's demand builds on word-of-mouth in the later weeks though rarely in practice.

$$\sum_{j=1}^{N}\sum_{s=1}^{H} x_{jsu} \le H \qquad \forall \quad u = 1..T \tag{4.4.3}$$

$$x_{jsu} = 0 \qquad \forall \quad j, s \text{ and } d_j < u < r_j \tag{4.4.4}$$

$$\sum_{s=1}^{H} x_{jsu+1} \le \quad \text{Max}\left\{\left(1 - \sum_{w=1}^{u} y_{jw}\right), \sum_{s=1}^{H} x_{jsu}\right\} \forall \ u=1..T-1, j \tag{4.4.5}$$

$$\sum_{w=1}^{u} y_{jw} = \sum_{v=1}^{u}\sum_{s=1}^{H} x_{jsv} \qquad \forall \quad j, u= 1..T \tag{4.4.6}$$

$$\sum_{j=1}^{N} x_{jsu} = 1 \qquad \forall \quad s, u \tag{4.4.7}$$

$$x_{jsu}, y_{jw} \in (0, 1) \tag{4.4.8}$$

In the above model, Constraint (4.4.2) ensures that in a particular week u any movie j can only be scheduled at most on one screen, that is, it cannot be scheduled on more then one screen in a given week. The inequality sign is to ensure that the movie may or may not be scheduled. For example if $x_{111}=1$ i.e. movie 1 is scheduled on Screen 1 in Week 1 then the Constraint (4.4.2) $x_{111}+x_{121}+x_{131}+x_{141}+x_{151}+x_{161} \le 1$ implies $x_{121}=x_{131}=x_{141}=x_{151}=x_{161}=0$ as these are binary variables and can take the values 0 or 1.

Constraint (4.4.3) restricts the total number of movies scheduled in a given week to the total number of screens in the theater. Constraint (4.4.4) simply ensures that the movie can not be scheduled before its release date and after its due date. In most instances the movie runs in a continuous manner and it is very rare that the movie runs for few weeks and then resumes after a break of one or more then one weeks.

Constraint (4.4.5) deals with the continuity of the movie run. It restricts the movie to play in breaks. So this constraint schedule the movie in such a manner that the movie run is continuous. There are two situations that may arise. We discuss these situations by taking a simple example:

Case 1: Movie plays for few weeks and then a break of one or more weeks

Suppose the release date of the Movie j is Week 1 and it runs in Week 1 and 2 and does not run in Week 3.

So $r_j = 1, y_{j1} = y_{j2} = 1$ and y_{j3}=0

So due to constraint (4.4.5),

$$x_{j14} + x_{j24} + x_{j34} + x_{j44} + x_{j54} + x_{j64} \leq \max\left\{\begin{matrix}(1-1-1-0), \\ x_{j13} + x_{j23} + x_{j33} + x_{j43} + x_{j53} + x_{j63}\end{matrix}\right\}$$

as movie is not played in Week 3, therefore,

$$x_{j13} + x_{j23} + x_{j33} + x_{j43} + x_{j53} + x_{j63} = 0$$

Hence,

$$x_{j14} + x_{j24} + x_{j34} + x_{j44} + x_{j54} + x_{j64} \leq Max \ \{\text{a negative number}, 0\}$$

$$x_{j14} + x_{j24} + x_{j34} + x_{j44} + x_{j54} + x_{j64} \leq 0$$

Hence the movie j cannot be scheduled in Week 4.

Case 2: Movie not scheduled on its release date

Suppose movie is not scheduled in Week 1, 2 and 3.

$r_j = 1, \ \ y_{j1} = y_{j2} = y_{j3} = 0$

therefore, according to constraint (4.4.5),

$$x_{j14} + x_{j24} + x_{j34} + x_{j44} + x_{j54} + x_{j64} \leq Max\left\{\begin{matrix}(1-0-0-0), \\ x_{j13} + x_{j23} + x_{j33} + x_{j43} + x_{j53} + x_{j63}\end{matrix}\right\}$$

As movie is not played in Week 3, therefore

$$x_{j13} + x_{j23} + x_{j33} + x_{j43} + x_{j53} + x_{j63} = 0$$

So $\quad x_{j14} + x_{j24} + x_{j34} + x_{j44} + x_{j54} + x_{j64} \leq Max\{1,0\}$

$$\leq \ 1$$

Hence the Movie j can be scheduled in Week 4.

However, Constraint (4.4.5), introduces nonlinearity into the system and hence, difficult to solve exactly. This justifies the use of a heuristic method for addressing this problem.

4.5 Genetic Algorithms

The concept of Genetic Algorithms (GA) was first proposed by Holland (1975). The basis for the algorithm was the observation that a combination of reproduction and natural selection allows nature to develop living species that are highly adapted to their environment. To apply GA to a problem, the candidate solutions have to be formulated in terms of strings.

Genetic Algorithm is basically a search algorithm in which we search for an optimal solution point for a certain objective. At each iteration, known as a *generation*, we have a fixed number of solution points. Each solution point is called as a *chromosome*. The fixed number of chromosomes at any generation is called as the *population size*. With each generation we move on to better and better quality of chromosomes, with respect to the objective decided. The maximum number of generations is decided by certain convergence criterion.

Initially we start with chromosomes that are generated either randomly or by certain heuristic. This group of initial chromosomes, which are equal in number to the population size, is called the *initial population*. Each chromosome is evaluated for the objective that has been decided. To move on to the next generation, we use certain *genetic operators*. We select some chromosomes from the current population, which are used for generation of the next generation. There are different ways to select these chromosomes. The basic objective of the selection operator is to increase the proportion of good chromosomes in the *mating pool*. A mating pool is a group of chromosomes, which will undergo certain operations to produce the next generation chromosomes which are better than the present ones, when evaluated against same criteria.

The genetic operators used to produce the next generation which has 'better' chromosomes, are *crossover* and *mutation*. Each of the chromosomes of the mating pool does not undergo mating. For the crossover operation, each time a pair of chromosomes is selected and with a certain prefixed probability, called *probability of crossover*, they are allowed to crossover. In crossover, the selected chromosomes exchange elements with each other. The underlying principle of crossover is that with

the exchange of elements between chromosomes we may get better ones. A crossover site in a chromosome, along which the elements are exchanged, is selected randomly. The mutation operator is used for diversification. If only crossover operator is used the search gets confined to certain space and solutions can converge at the local optima. In mutation we allow the chromosome to diversify from its parent. The mutation operator is generally applied bit-wise, that is, we have a pre-defined probability, called *mutation probability*, with which an element in the parent chromosome is allowed to mutate. The element takes a diverse value at random and thus results in the diversification of the population. The approach therefore operates in the following manner.

4.5.1 Algorithm Process

The basic GA process can be visualized as follows (Michalewicz 1992, Grefenstette 1986). Here POP(t) indicates population at generation t.

Step 1: $t \leftarrow 0$

Step 2: Generate initial population, POP (t) either randomly or using a heuristic.

Step 3: Evaluate each of the strings in POP(t).

Step 4: IF stopping condition satisfied THEN stop ELSE continue.

Step 5: $t \leftarrow t + 1$.

Step 6: Select strings from POP(t -1) to create POP(t).

Step 7: Apply genetic operators on strings in POP (t).

Step 8: Evaluate each of the strings in POP(t).

Step 9: GO TO 4.

The algorithm thus works on the fact that a "good" chromosome contains some features (sub-strings) that are desirable and hence contribute to its being evaluated high. With the exchange of genetic material between two good strings, the expectation is that the offspring generated will be good. Thus, as the iterative process continues, we move to better and better population of solutions.

GAs have been applied in a variety of fields. Goldberg (1989) has used GA for optimization of pipeline systems; Cleveland and Smith (1989) for scheduling flow shop releases; Liepins and Potter (1991) use GA for multiple fault diagnosis;

Balakrishnan and Jacob (1996) use GA for product design; Deb, Chakraborty and De (1999) use GA for model based object recognition; Joshi (2000) uses GA in search space of hypercubes.

4.5.2 GA in Multiple Screens Problem

By the application of Genetic Algorithm to solve the multiple-screen problem, we develop an approach for finding a schedule to maximize the total revenue the exhibitor will obtain.

For the multiple-screen problem, GA views sequences of movies to be displayed on screens for each week, for the total planning horizon, as a chromosome (the candidate solution), which in turn is a member of a population. Fitness of each chromosome is measured by the value of the objective function (i.e. the total revenue earned by the exhibitor.) The GA performance depends greatly on the choices of the different GA parameters, population size (p_s), the probability of crossover (p_c) and the probability of mutation (p_m). Selection of these parameters is itself an optimization problem and these parameterization have also been noted to be problem specific (Kumar, Bagchi and Sriskandarajah, 2000; Davis, 1991; Grefenstette, 1986;). We use a design of experiments (DOE) approach (Bagchi and Deb, 1996) to quickly identify the good settings for p_s, p_c, and p_m before using GA to solve our problem.

We define an n x w vector to represent a chromosome for an n screen w week problem. For example, a chromosome for a two-screen two-week problem can be represented as $m_1 \quad m_5 \quad m_1 \quad m_3$. The chromosome represents that Movie m_1 is played in Weeks 1 and 2 on Screen 1, Movie m_5 is played in Week 1 on Screen 2 and Movie m_3 is played in Week 2 on Screen 2. The GA begins with a population (population size p_s) of randomly generated vectors. The fitness of any chromosome is then evaluated based on the total revenue for that chromosome. To calculate the total revenue corresponding to a movie when placed in a particular week-screen slot, the number of visitors is taken as the minimum of the respective screen capacity and demand of the movie in that week. Using this number and the contract term, we calculate the revenue earned by that movie in that week. Total revenue is the cumulative of all such week-screen slot corresponding revenues.

screen slot corresponding revenues. The revenue that the exhibitor earns from a week u screen s slot if movie j is scheduled in this slot is given by

$$\mathrm{Re}venue_{jsu} = [Min\{cap_s, demand_{ju}\}] * (1 - floor_{j,r})$$

where $floor_{j,r}$ is percentage share of distributor for movie j, when run for weeks r and cap_s is capacity of screen s and $demand_{ju}$ is the demand for movie j in week u.

4.5.3 GA Parameters

We now briefly provide overview of Genetic Algorithm parameters employed in the current study that are particularly germane to the multiple-screen problem. We begin by discussing the operationalization of GA operators to the specific class of chromosomes defined above.

Selection: We use the method of tournament selection. In this type of selection, at a time two chromosomes are selected randomly from the current population and compared on the basis of their fitness value (total revenue). The one with high fitness value is added to the mating pool. This procedure is repeated for p_s number of times, and each time the winner is added to the mating pool. Thus we maintain the total population size p_s in the next generation. Since the selection is based on the fitness value the 'good' chromosomes have a higher probability of getting selected for the mating pool as compared to bad ones. This increases the proportion of good chromosomes in the mating pool. These selected members are given a chance to mate according to the probability of crossover (p_c).

Crossover: A key consideration in designing the crossover operator is avoidance of infeasible solutions. First, we select the mating pairs of chromosomes at random from the mating pool. Then with a probability p_c we allow them for mating. Then, we use "one-point" crossover (Murata and Ishibuchi, 1994) in which one point is randomly selected for dividing one parent. This point is the crossover site. Due to the complex structure of the chromosome, we restrict the crossover site to be a starting point of any week. Thus the crossover site can be either starting point of the first week or second week and so on. This avoids the in-feasibility of resulting chromosome due to release date constraint. The elements on the left-hand side of the crossover site are transferred from the parent to one child. The elements of right hand side for the child are the

Figure 4.1(a) Parent Chromosomes before Crossover

Parent 1

Screen

Week	1	2	3	
1	m_1	m_3	m_5	
2	m_3	m_5	m_4	Crossover site
3	m_3	m_5	m_2	
4	m_2	m_6	m_5	

Parent 2

Screen

Week	1	2	3	
1	m_7	m_8	m_9	
2	m_8	m_{10}	m_9	Crossover site
3	m_8	m_{11}	m_9	
4	m_{11}	m_{12}	m_8	

With crossover point as starting of third week, the children formed will be as shown in Figure 4.1(b).

Figure 4.1(b) Children Chromosomes after Crossover

Child 1

Screen

		1	2	3
Week	1	m_1	m_3	m_5
	2	m_3	m_5	m_4
	3	m_8	m_{11}	m_9
	4	m_{11}	m_{12}	m_8

Child 2

Screen

		1	2	3
Week	1	m_7	m_8	m_9
	2	m_8	m_{10}	m_9
	3	m_3	m_5	m_2
	4	m_2	m_6	m_5

Figure 4.2 Mutation: Replacement by a Movie Not Previously Scheduled.

Chromosome Before Mutation of Element Week3-Screen2 (i.e. m_3)

		Screen 1	Screen 2	Screen 3
Week	1	m_7	m_8	m_9
	2	m_8	m_{10}	m_9
	3	m_3	m_5	m_2
	4	m_2	m_6	m_5

Chromosome After Mutation of Element Week3-Screen2

		Screen 1	Screen 2	Screen 3
Week	1	m_7	m_8	m_9
	2	m_8	m_{10}	m_9
	3	m_{27}	m_5	m_2
	4	m_2	m_6	m_5

Figure 4.3 Illustration of use of Repairing Function

Infeasible Chromosome due to Discontinuity of movies m_5 and m_{16}

		Screen		
		1	2	3
Week	1	m_7	m_5	m_{16}
	2	m_8	m_{10}	m_9
	3	m_3	m_5	m_2
	4	m_2	m_6	m_5
	5	m_2	m_5	m_{16}

Infeasible Chromosome made Feasible After Repairing Function

		Screen		
		1	2	3
Week	1	m_7	m_5	m_{16}
	2	m_8	m_5	m_9
	3	m_3	m_5	m_2
	4	m_2	m_6	m_5
	5	m_2	m_5	m_4

4.5.4 Design of Experiments: Parameter Optimization Approach

The GA parameters, namely, the population size p_s, the probability of crossover p_c, and the probability of mutation p_m, have a great impact on the performance of the GA

based heuristic. Here we explain the *Design of Experiments* approach to optimize the three GA parameters. Each of these parameters has two settings for which the performance of GA is tested. The experimental setup is as shown in Table 4.1.

There are eight experiments that cover all combinations of settings for these three factors. The '0' setting for a factor indicates that it is set to a lower amongst the two values, and the '1' setting indicates that it is set to a higher value. For each experiment, the GA is run for a fixed number of generations and the revenue corresponding to the best schedule is noted down. Then for a factor, the sum of revenues corresponding to all the experiments in which the factor was set at '0' is compared with that of experiments where it was set to '1'. The setting of the factor with higher value for the sum is then considered better. This is done for all the factors. Now if setting '1' is proved to be better for a factor, we make it as a '0' setting and make some higher value as a '1' setting for the factor, thus moving in the direction of higher values. Similarly, if '0' setting proves better, we move in the lower direction for that particular factor. With these new settings the process is repeated. If moving in any direction from a setting for a factor degrades the performance of GA, we conclude the setting for the factor as optimal.

Table 4.1 Experimental Setup for GA Parameter Optimization

Experiment	p_s	p_c	p_m
1	0	0	0
2	0	0	1
3	0	1	0
4	0	1	1
5	1	0	0
6	1	0	1
7	1	1	0
8	1	1	1

4.6 Performance Evaluation

Saxena (2000) reports that the integer programming model based heuristic worked quite well for multiple-screen problem. However, as illustrated in the illustrative example of Section 4.3.2.4, his approach may lead to inferior solutions since first the movies to be scheduled are selected by the integer programming model, without considering the screen capacities and then the screens are allotted to the movies by considering only their demand. In allocating the screens to movies an important factor of contract term between the exhibitor and distributor is neglected.

The GA approach that we propose is a search method, and hence may lead to actual optimal solution. To demonstrate the effectiveness of our approach, we need to show that under certain conditions of the input parameters, a GA-based heuristic, which explicitly considers the capacity variations in the screens works better.

We now explain the methodology to compare the performance of the proposed GA-based heuristic with the Saxena's (2000) heuristic. The problems examined for this purpose were generated in a similar manner of the previous chapter. Movies are classified into four "types" based on their opening strength and decay rate. As explained in the implementation section of the MDP model in previous chapter, the data of 84th St. Sixplex, a six-screen theater in New York, was used to estimate the opening strength and the decay rate of each of these types of movies. We considered 27 weeks of the year 1989 and a corresponding period from its preceding year 1988. We classified the movies of the two years separately in movie types following Jedidi, Krider, and Weinberg (1998). First, data were median split on the basis of the opening strength of movies. Then, each of these groups was median split on the basis of their decay rates. The median values of the data for each of the four groups were used as the representative opening strength and decay rates for the groups. We estimated the demand pattern for each type of movie by using the equation $5 * e^{\alpha - \beta t}$ where α represents the opening strength, β represents the decay rate, and t represents the number of weeks after the release of movie. The constant multiplier 5 was chosen arbitrarily as we were interested in getting only the representative demand curve for each type of movie and not the actual demand for these movies types.

We first discuss the factors that we considered to be significant in deciding the improvement of GA-based heuristic over Saxena's (2000) heuristic. The constraint of capacity of screens is the extra constraint that is considered by the GA-based heuristic. Therefore, we considered *screen capacity* to be one of the significant factors. Second, we found in our preliminary analysis that SilverScreener has a tendency to select *Type I movies*[4] in their preliminary weeks because of their high opening strength. However with the added constraint of capacity, the value addition in the objective function corresponding to Type I movies might be reduced. Therefore, we select the number of Type I movies in a scenario as another factor to be analyzed. The third factor we analyze is the *contract term* between the exhibitor and the distributor, which decides the actual revenue earning strength of the movies for the exhibitor. To summarize we study the impact of the variation in the capacity of screens, the number of Type I movies released and the contract term between the exhibitor and the distributor, on the improvement shown by GA-based heuristic over Saxena's (2000) heuristic.

We examine two levels, high and low, of the *capacity* factor. To consider the typical distribution of capacity over screens, we use the screen capacity data of a theater in The Netherlands. In the high capacity case, we assume eight shows in each screen room per week and in low capacity we assume four shows per week[5]. Table 4.2 gives the capacities we assumed for each of the cases[6].

Table 4.2 Capacity of Screens

	Capacity of Screen					
Case	1	2	3	4	5	6
High Capacity	3472	2736	1728	1208	1112	904
Low Capacity	1736	1368	864	604	556	452

[4] Type I movies are the movies with high opening strength but decay rapidly.
[5] Again these numbers of shows do not exactly represent the actual number of shows in a theater, and are scaled according to demand of movies in initial weeks.
[6] Note that in the two levels of Capacity we do not change the proportion of distribution of capacity over screens. Studying the impact of change in this proportion can be an interesting study and recommended for future research.

To operationalize the proportion of Type I movies in a scenario, we classify the scenario as a *high 'decay property'* scenario in which there are ten or more rapidly decaying movies (Type I). Similarly, if there are three or less number of Type I movies, we classify the scenario as *low 'decay property'* scenario.

We consider the performance of GA under two types of contract terms that could be used, namely, first a 90%/10% contract term (common for all movies) and second movie type based contract terms.

The 90%/10% contract term is used with the minimum percentage share as explained in the Section 4.3.1, with a house nut of 600 (scaled as per other factors, Squire, 1992). In their study of contract terms Swami, Lee and Weinberg (2000) suggest that movie type based contract terms result in improved channel coordination and hence recommend the same to the distributors. A typical movie based contract term considered by us is shown in Table 4.3.

Table 4.3 Movie based Contract term

	Exhibitor's % Share in Total Revenue for Movie Type			
Week	I	II	III	IV
1	10	15	25	25
2	30	20	40	40
3 onwards	50	35	50	50

These are according to the industry practice. Movies of Type I open with high demand and decay rapidly, so distributors of these movies have a tendency of retaining maximum share of revenue in their initial weeks. Since Type II movies retain their strength for long, the distributor offers less share of revenue from these movies to the exhibitors. Movies of Type III and IV open with low strength and never have less demand, so to promote exhibitors to play these movies, distributors offer more share in the revenue for such movies.

The input parameters required to generate problems for various simulation analyses are as follows:

i. length of planning horizon
ii. number of screens at the theatre
iii. quantity of movies (i.e., total number of movies considered) in a scenario
iv. quality of movies
v. release dates of movies

To generate different simulation problems, we assign fixed values to the first three parameters, namely, the length of planning horizon, the number of screens in a theater and the quantity of movies released during the season, which are common to all problems. We set the length of the planning horizon to be 8 weeks (e.g., representing the two peak months of summer season) and the number of screens to be 6. The quantity of movies released is also kept constant. We assume release of 4 movies in each week. To begin with we assume that there are six movies already playing on the six screens at the end of the previous season and that the exhibitor can continue to play any of them in the new season.[7] Therefore the movies released in week one of the season are not guaranteed of exhibition unless the exhibitor finds it more profitable to replace some of the currently playing movies with the newly released movies. The release dates of movies are thus fixed by assigning the number of movies released per week to 4.

To randomize the assignment of a movie to a movie type, we use the relative proportions of Type I (19%), II (7%), III (38%), and IV (36%) movies in Jedidi, Krider, and Weinberg's (1998) sample. Specifically, a random number is drawn among integers between 1 and 100 for each movie. Then the movie is assumed to be Type I if the random number is between 1 and 19, Type II if it's between 20 and 26, Type III if it's between 27 and 64, and Type IV if it is between 65 and 100.

4.6.1 Simulation Setup

We study the effect of capacity of the screens (high/low), type of contract term (90%/10%/movie based), and decay property (high/low) for the period, on the

[7] Thus there are 38 movies in each simulated season.

difference between exhibitor's cumulative revenue from proposed heuristic and Saxena's (2000) heuristic. We use a complete 3 (treatment) X 2 (level) factorial design. Movie based contract term is denoted by type 1 and 90%/10% contract term is denoted by type 2.

4.6.2 Methodology

Using the above scheme of simulating problems, a total of 58 problems were generated. Based on the input data, the problems were assigned to their respective type of experiment. The GA model of Section 4.5 was coded in C++ language (Appendix 2) and run on an Intel Pentium III class computer. The value evaluations were done according to the cumulative revenue criterion presented in Section 4.5.2.

4.6.3 Simulation Results

Table 4.4 summarizes the simulation results in several ways. Table 4.4(a) gives the mean (and standard deviation) of improvement shown by GA-based heuristic over Saxena's (2000) heuristic, under each factor level combination and Table 4.4(b) gives the mean and standard deviation of improvement shown by GA-based heuristic over Saxena's (2000) heuristic, for each main effect. The ANOVA results in Table 4.4(c) show that two interactions are not significant and that all three main effects (capacity, decay property and contract term) and an interaction of capacity and contract term are significant at the *.05* level.

Table 4.4 (a): Mean Improvement in Cumulative Revenues by GA-based Heuristic over Saxena's (2000) Heuristic for Experimental Setup

Exp.	Capacity[a]	Contract term[b]	Decay Property[c]	Number of problems	Mean improvement in cumulative revenue*	% improvement
1	0	0	0	6	4354(1183)	42
2	1	0	0	6	3230(737)	21
3	0	1	0	6	2488 (544)	22
4	0	0	1	6	5859 (449)	56
5	1	1	0	6	2181(843)	13
6	0	1	1	6	3695(551)	35
7	1	0	1	6	3595 (511)	21
8	1	1	1	6	3609 (397)	23
				Total = 48**		

a – High (1) setting implies high capacity levels, otherwise low (0) setting
b – Setting 0 implies movie based contract term, setting 1 implies 90/10 contract term
c – High (1) setting implies more than 9 type I movies in the scenario, Low (0) setting implies less than 4 type I movies.
* Standard deviation values are shown in parentheses.
** The total number of problems is 48 as 10 out of the randomly generated 58 problems could not be assigned exactly to an experiment type because the number of type I movies was in between 3 and 9.

Table 4.4 (b): Mean Improvement in Cumulative Revenues by GA-based Heuristic over Saxena's (2000) Heuristic for Different Settings of Parameters

Factor	Setting	Mean Improvement in Cumulative Revenue*
Capacity	High	3154 (622)
	Low	4099 (681)
Contract term	90%/10%	2993 (583)
	Movie based	4260 (720)
Decay Property	High	4190 (477)
	Low	3083 (826)

* Standard deviation values are shown in parentheses

Table 4.4 (c): ANOVA Results for Improvement of GA-based Heuristic over Saxena's (2000) Heuristic

Source of Variation	Sum of squares	Degrees of freedom	Mean square	F_0
Capacity (A)	10729534	1	10729534	6.95*
Contract term (B)	19243201	1	19243201	12.42*
Decay Property (C)	15223521	1	15223521	9.86*
AB	6727518	1	6727518	4.35*
AC	632502	1	632502	0.4
BC	438536	1	438536	0.28
ABC	1386520	1	1386520	0.89
Error	18172536	40	454313	
Total	72553868	47		

- Significant at 5 percent.

4.6.4 Discussion

Intuition suggests that the factor of capacity should be the most important factor in improvement of cumulative revenue by GA-based heuristic as compared to Saxena's

(2000) heuristic. The reason for this argument is that, the only difference in the Saxena's (2000) heuristic and the proposed GA-based heuristic is that of a capacity constraint. However, our results show that other factors such as contract term and decay property of scenario are also important. In fact, the contract term is most significant factor followed by the factor of decay property and the capacity factor. The only interaction that is significant is the capacity and contract term interaction. In the 90%/10% contract term, the exhibitor gets more or less an equal percentage of share for all types of movies. Therefore, in case of 90%/10% contract term, the only factor which remains important in optimal selection of movies is the demand for a movie. If the capacity constraint is applied in such situations, the Screen Allotment heuristic of Saxena (2000) might also provide good solutions. When the movie based contract term is applied, a movie of Type I which has a high demand at opening but, a low percentage of share for exhibitor in the initial weeks, does not remain optimal after the application of the capacity constraint. This is because the capacity constraint limits the demand for any type of movie to a maximum of screen capacity, while the percentage share of exhibitor in the total revenue, is high for other types of movies. Thus the factor of contract term between the exhibitor and distributor is most important factor.

The second factor found important is the decay property of a season or the total number of rapidly decaying movies in a season. It is important to note that the setting for this factor were very extreme settings. This may have introduced an artificial significance to this factor. Since Type I are the rapidly decaying movies the distributor of such movies have the tendency of retaining maximum revenue, from the first few weeks exhibition of these movies. So the initial exhibitor share for such movies is less. Each movie of Type I scheduled by Saxena's (2000) heuristic leads to degradation of the schedule, because first, the exhibitor share in profit is less and second, these movies are allotted the screens with maximum capacities because of their high demand. Therefore, the number of such movies in a season affects the improvement shown by GA-based heuristic over Saxena's (2000) heuristic.

The only interaction that was significant was the capacity and contract term interaction. The proposed GA-based heuristic shows more improvement over

Saxena's (2000) heuristic if this interaction is high. When the capacity is set at low value (0) and the contract term factor is also set at movie based contract term (0), then their interaction is high. This combination of settings has a very large improvement shown by GA-based heuristic over Saxena's (2000) heuristic, than any other setting combination of these factors. In combinations like high capacities and a movie based contract term, the effect of contract term is offset by the relatively loose capacity constraint. Similarly, with the 90%/10% contract term, the effect of low capacity constraint is offset.

4.7 Managerial Implications, Conclusion and Directions for Future Research

The proposed GA-based model outperforms the non-capacity based heuristic by 25% on an average. This suggests the importance of considering the screen capacities in the exhibitor's decision making. All the main factors, namely, the capacity levels, the type of contract term, and the number of rapidly decaying movies, are significant. The research suggests some important policies for the exhibitors as well as the distributors of movies. First, it is implied that the exhibitors should not allocate screens to movies based on their demand alone. Even if the movies with high opening strength has higher demand in the beginning, its allocation to the high capacity screens might lead to inferior schedules because of the contract terms specific to movie types. The interaction of factors of capacity and contract term is significant. This suggests that the exhibitors with low capacity screens should not blindly exhibit movies of Type I. They must take into consideration other factors such as contract term.

On the other hand, distributors of Type I movies should go for 90%/10% contract term with the exhibitors having low capacity screens while a movie based contract term with exhibitors having high capacities. Also the exhibitors with higher capacity screens should exhibit Type I movies only at screens with capacities more than the demand of other movies if the contract terms are movie based. Exhibitors with low capacity screens should prefer movie based contract terms with the distributors distributing less number of Type I movies and should prefer 90%/10% contract terms with distributors distributing more number of Type I movies.

In the current research we have considered only one theater. However this model can be extended to consider an exhibitor who owns multiple theaters. In fact this is the common practice, and in any regional area, typically most theater screens are controlled by a few chains. Such a setting may allow the theater owner to optimize over multiple sites as the decision of scheduling movies on a theaters are highly influenced by the other theater.

The current research can also be extended to address the competitive issues. We consider the movies as independent entities and the performance of one movie is not affecting the demand of other movie. However in the real world situation the performance of one movie may affect the others. Consideration of such effects would provide interesting future research ideas.

CHAPTER 5

MODEL EXTENSIONS IN OTHER APPLICATION AREAS

In this chapter we discuss the extensions of SilverScreener in addressing some problems in similar areas such as television programs scheduling and retail space allocations. We also discuss some of the related areas in which meta heuristics such as GA can used.

5.1 Television Program Scheduling

In their research, Reddy, Aronson and Stam (1998) propose a model S.P.O.T. (Scheduling Programs Optimally for Television) which explains a network analogy with scheduling of programs on television. The model considers an interesting issue of "lead-in" s of television programs. They define lead-in of a program as its strength to boost the ratings of a newly introduced or a weaker program. With a minor change in model, SilverScreener can also take into consideration this "lead-in" effect and can be efficiently used in television programs scheduling, with an advantage of simplicity over S.P.O.T. We now explain a model, which is based on SilverScreener and is equivalent to the S.P.O.T. model. We then extend this model for addressing some issues, which are not addressed by S.P.O.T.

S.P.O.T. equivalent model

We first define the terms to be used in the model:

N : Total number of television programs available for scheduling.

W : Number of uniform time slots to be scheduled.

LI_k : lead-in effect on next scheduled program due to program k.

l_j : length of program j in number of uniform time slots.

y_{jkw} : binary decision variable

$y_{jkw} = 1$ if program j follows program k, and begins at slot w.

$= 0$ otherwise.

$x_{j,w}$ = binary decision variable

$x_{j,w} = 1$ if program j is scheduled on slot w.

$P_{j,w}$ = contribution to objective function if program j is scheduled at slot w.

The S.P.O.T. equivalent model will be as follows:

$$\max \sum_{j=1}^{N}\sum_{w=1}^{W} P_{jw} * x_{jw} + \sum_{w=1}^{W}\sum_{j=1}^{N}\sum_{k=1}^{N} LI_k * y_{jkw} * P_{jw}$$

Subject to

$$\sum_{j=1}^{N}\sum_{w=l_j+1}^{W} x_{ji} \leq 1 \qquad \forall\, w = 1......W \qquad ....(5.1)$$

$$y_{jkw} \leq (x_{jw} + x_{k(w-l_k)})/2 \qquad ....(5.2)$$

$$x_{jw} \in (0,1) \qquad \forall\, j = 1........N, \quad \forall\, w = 1....W \qquad ....(5.3)$$

$$y_{jkw} \in (0,1) \qquad \forall\, j = 1..N, \quad \forall\, w = 1....W, \quad \forall\, k = 1....N \qquad ....(5.4)$$

$$x_{jw} \geq 0 \qquad \forall\, j = 1........N, \quad \forall\, w = 1....W \qquad ....(5.5)$$

$$y_{jkw} \geq 0 \qquad \forall\, j = 1..N, \quad \forall\, w = 1....W, \quad \forall\, k = 1....N \qquad ....(5.6)$$

Constraint (5.1) checks that only one show is scheduled at any point in time. Constraint (5.2) takes into account the lead-in effect of show k, only if it is scheduled just before show j.

To illustrate the model developed, we explain a simple example. Suppose that there are 10 programs to be scheduled and we have a total of 6 slots. Each program has the play-length of 2 slots. Program number 7 has a lead-in effect and is scheduled in slot number 2. Thus $x_{72}=1$; now if we have to schedule program 3 at slot 1 then $x_{31}=1$.

The constraint (5.1) will give for slot 2,

$x_{31} + x_{32} + ...+ x_{71} + x_{72}$ <=1

i.e. 1+0+.........0+0+1<= 1. which is not satisfied. Thus Constraint (5.1) will limit the number of programs to be scheduled at any slot to 1.

Now if program 3 is scheduled at slot 5, we have x_{35}=1. Then by constraint (5.2), (5.4) and (5.6), we have $y_{375} <= (x_{35} + x_{73})/2$ i.e. <= (1+0)/2 = 0. Thus as program 3 did not immediately follow the program 7, the lead-in effect of program 7 will not be considered. This way both the constraints work well and the above model is equivalent to the S.P.O.T. model.

We can add one more practical consideration into the model. The S.P.O.T. model does not incorporate daily shows. We present a model that considers the telecasting of programs that are shown daily throughout the week. We define

P_{jwi}: Contribution of show j towards objective function, when scheduled at time slot w of day i.

x_{jwi} : Binary decision variable takes value 1, if show j is scheduled at time slot w of day i., 0 otherwise.

B_{jw} : Contribution of show j towards objective function, when scheduled at time slot w for the whole week.

Y_{jkwi} : Binary variable, takes value 1, if show j is scheduled at time slot w on day i, and show k is scheduled just before that, 0 otherwise.

Z_{jw} : Binary variable, takes value 1, if show j is scheduled at time slot w for the whole week, 0 otherwise.

The model will be:

$$\max \quad \sum_{j=1}^{N}\sum_{w=1}^{W}\sum_{i=1}^{7} P_{jwi} * x_{jwi} + \sum_{j=1}^{N}\sum_{k=1}^{N}\sum_{w=1}^{W}\sum_{i=1}^{7} LI_k * y_{jkwi} + \sum_{j=1}^{N}\sum_{w=1}^{W} B_{jw} * z_{jw}$$

Subject to

$$y_{jkwi} \le (x_{jkwi} + x_{kiw-l_k} + z_{jw} + z_{k(w-l_k)})/2 \quad \forall j,k,w,i \quad \text{.......... ....(5.7)}$$

$$\sum_{j=1}^{N} \sum_{r=w-l_j+1}^{w} x_{jri} + \sum_{j=1}^{N} \sum_{r=w-l_j+1}^{w} z_{jr} \le 1 \quad ...\forall\ i = 1....7, w = 1....W \quad \text{.......(5. 8)}$$

$$\sum_{j=1}^{N} \sum_{i=1}^{7} x_{jwi} + 7 * \sum_{j=1}^{N} z_{jw} \le 7...... \forall w = 1.....W \quad \text{.......... ......(5.9)}$$

$x_{jwi} \in (0,1),\quad x_{jwi} \le 0,$ ....$\forall j = 1...N,\ \forall w = 1.....W,\quad \forall i = 1....7$ ....(5.10)

$y_{jkwi} \in (0,1),\quad y_{jkwi} \le 0$ ....$\forall j = 1...N,\ \forall w = 1.....W,\quad \forall i = 1....7\quad \forall k = 1...N$

...(5.11)

$z_{jw} \in (0,1),\quad z_{jw} \le 0,$ ....$\forall j = 1...N,\ \forall w = 1.....W$......................(5.12)

Constraint (5.7) ensures that lead-in effect is considered only when show j follows show k. Constraint (5.8) ensures that at one time slot on a particular day only one show is scheduled. Constraint (5.9) insures that over a week at a particular time slot only maximum of 7 shows are scheduled.

We illustrate the above model with an example. Suppose there are 10 slots per day to be scheduled and there are 30 programs with a play length of 3 slots each. If program 7 is scheduled for a week on slot 6 then z_{76}=1. Now if we schedule program 3 on slot 7 of the day 4 then x_{374}=1.

Constraint (5.8) gives

x_{374} + x_{364}...+ x_{354}.........+z_{76} < = 0. i.e. 1+....0.+1< = 0, which is not satisfied.

Also by constraint (5.7) the lead-in effect of program 7 will come into effect for program 3 only when either: a) program 7 is scheduled exactly before program 3 on one day, or b) program 7 is scheduled as a daily program and program 3 follows program 7 on one day, or c) program 3 is scheduled as a daily show and program 7 precedes program 3 on one day, or d) both programs are scheduled as daily shows and program 7 precedes program 3 on all 7 days.

This model still does not consider a show's re-telecasting possibility. If we consider re-telecasting, the model will be nonlinear as each re-telecasting will affect

the contribution of show towards the objective function. This extension will thus require a heuristic method such as the one explained in previous chapter. The proposed Genetic Algorithm heuristic can directly be used for solving this complex problem since with the possibility of re-telecasting, it can also consider the deterioration of contribution to the objective function value with each re-telecast.

5.2 Retail Space Management Considering Demand Elasticities

Corstjens and Doyle (1983) explain a model for retail space management. Their model considers elasticity in demand (sales) of a product with varying amount of space allocation to it. They also consider a cross elasticity in demand of products. That is, they consider the effect of certain space allocation of one product to the sales of another product. Such conditions are interesting and GA can be applied to such problems, which will help obtain near optimal solution. For the similar problem of retail space management, Boris, Farris, and Freeland (1994) explained the use of another meta heuristic, Simulated Annealing, after modelling the problem as a constrained optimization problem.

5.3 Internet Space Scheduling

The number of World Wide Web users is increasing at a rapid rate. This has made Internet an important media for advertisement. The scheduling of adds on the web site poses some interesting problems in front of web site administrators. The pricing is based on the site exposure or the banner ad exposure. An advertisement on the web has two parameters, namely the size of advertisement and its display frequency. To schedule ads on the web in a way as to optimize on both the parameters is a difficult task. Kumar, Jacob, and Srikandarajah (2000) use GA in this area. They propose an algorithm based on GA, which they call hybrid-GA that gives significant increase in objective function values as compared to other existing algorithms for this problem.

This problem of advertisement scheduling is analogous to the retail space allocation problem discussed earlier. The site administrator has certain space, which is to be allocated with an objective of maximizing the revenue earned. At each decision epoch, the administrator takes a decision of which advertisements to schedule and

what space to be allocated to each advertisement. Depending upon the action taken by the administrator, the system makes transition to a new state. This state can be defined as the function of the advertisements scheduled and the respective space allocations. A probability distribution of revenue that will be generated is associated with each state. Thus the Internet ad scheduling problem is in one way a combination of the two retail space allocation problems we have considered, which has a complexity in terms of varying space allocation to each advertisement as well as stochasticity of revenue corresponding to each allocation.

CHAPTER 6

CONCLUSION AND LIMITATIONS

In this thesis we applied marketing and management science (operations research) methods along with a tool of Genetic Algorithm to dynamic retail management of movies. We showed how quantitative model building and meta-heuristics like Genetic Algorithms can assist marketing decision making in complex environments involving perishable products. We addressed the general retail space allocation problem, in complex situations such as stochasticity in the demand of products and the display of multiple products. This space allocation problem was studied in the context of motion picture exhibitors.

We described the exhibitor problem in two ways. In the first problem the exhibitor has a single screen and the demand distribution for movies is considered to be stochastic. We propose an MDP based model to address this problem. In the second problem we consider an exhibitor with multiple screens and assume deterministic demand. We propose a GA-based heuristic to handle these problems. We explain how the models might assist the movie exhibitors in making profitable scheduling decisions. At a broad level, the model would benefit managers in two ways. One is by helping managers improve their scheduling techniques. That is, the model could help the manager adopt a new decision making style which is based on an analytical approach or a tested heuristic. The second is by helping the managers perform the same scheduling task in a more efficient way, using computer-based routines.

Although we have considered movies in this research, our results can readily be applied to any other perishable entertainment product. Thus in a broad sense, the current research addresses the problem of general retail space management. We also present extensions of the SilverScreener for its application in some other similar areas. We propose GA-based heuristics for addressing more complex situations, which cannot be efficiently tackled by simple models.

The current research has some limitations that suggest attractive ideas for future research. We address a single-screen problem with stochasticity in demand, and also a multiple-screen problem considering deterministic demand pattern for the movies. The problem of multiple-screens considering stochasticity in demand which will be very close to a practical situation, is yet unanswered. We believe that this problem can be answered efficiently a combinations of heuristic methods including GA.

There are computational limitations to our problem. The MDP code developed has limitations for the size of problems handled. We used certain state reduction techniques to solve the problem of sixteen movies in a scenario. The time take by this code for a 16-movie problem is about 45 minutes, which is very large. For the application of this model in the real world, it should be able to handle larger problems. A user friendly Decision support system based on these models can be developed.

To summarize, we present application of MDP approach in the area of retail space management, and the use of GA approach in the field of marketing. The thesis attempts to bring together the two important fields of marketing and management science/operations management.

APPENDIX 1

MDP CODE IN C LANGAUGE

```
// This program uses 3 ranks and for 16 movies. this program uses
single link list
#include<stdio.h>
#include "prmtrsom16.h"
// defining structure for a state
struct state
            {
                    int stag;
                    int indl;
                    int lev;
                    int rumovrank;
                    float maxm;
                    int max_ind;
                    struct state *fptr;
            };

void readata();  // function which reads data
void mat_gen();
//generated matrix of 3x8 for all combinations of 1,2,3 for 8 movies.
void conset();// generates consideration set foreach stage.
int check_state_feasibility(int mov,int lev);
float get_prb(int l,int lev1,int stage,int action,int runmv,int
act,int mov1);  // gives probability of transition from one state to
other
main()
{
        struct state *list,*list1,*current;
        int
stage,i,mov,run,lev1,j,act,c,l,action,t,pos1,m,k,runmv,flag,count,
            reduce;
        float MAXIM,tran_prb,x,exp=0;
        FILE *fout;
        fout=fopen("rescon8.dat","w"); //output file.
        list=(struct state *)malloc(sizeof(struct state));//this will
be the header for link list
        readata();//reading data from file
        mat_gen();// generating combinations of ranks from 1 to 4 and
storing in array level[NUM_LEVELS][NUM_MOVIES]
        conset();// generating a consideration set for each stage and
storing in conset[NUM_WEEKS][4]
        for(i=1;i<=NUM_MOVIES/2;i++)con_set[0][i]=con_set[1][i];
        current=list;//current record is initially the header;
for(stage=NUM_WEEKS;stage>=1;stage--)
{
 //loop for all currently running movies which are in the
consideration set of stage-1;
for(i=1;i<=NUM_MOVIES/2;i++)
    {
        mov=con_set[stage-1][i];// considering a current movie, which
will be one of the movies in action set for previous week.
        if(mov!=0)
```

```
        {
        for(run=0;run<=stage-rel_dt[mov];run++)//for each possible
run-length
            {
             for(k=1;k<=3;k++)//for each possible rank
             {
              flag=1;pos1=0;
              for(m=1;m<=NUM_MOVIES/2;m++)
              if(con_set[stage][m]==mov){flag=0;pos1=m;}
       for(l=1;l<=NUM_LEVELS;l++)
         {
if((flag==0)&&(level[l][pos1]!=k))continue;
if(k!=4)   //checks the feasibility of the state
{
current->lev=l;current->ind1=mov*10+run;current->maxm=0;
current->max_ind=30;current->stag=stage;current->rumovrank=k;
for(act=1;act<=NUM_MOVIES/2;act++)
 {
  MAXIM=0;
  action=con_set[stage][act];//movie to be considered as action
  if(action!=0)
      {

           c=level[l][act];// current rank for movie selected as
action.
      if(action==mov)t=run;// if action is currently playing movie
      else t=0;
      MAXIM+=rev[c][t];
      if(stage!=NUM_WEEKS)
         { // freeing records for stage next to next since they are
no more required.
         if(stage<NUM_WEEKS - 1)
         while(list->stag>stage+1)
           {
            list1=list->fptr;
            free(list);
            list=list1;
           }
            list1=list;
// moving state by state in records of next stage to calculate
revenue*transition probability, which will sum up to revenue
corresponding to this action.
            while(list1->stag!=stage)
             {
             if(list1->ind1==action*10+t+1)
             {
             lev1=list1->lev;

             runmv=list1->rumovrank;
             tran_prb=get_prb(l,lev1,stage,action,runmv,act,mov);
//getting probability of transition from current level to next level.
             MAXIM+=tran_prb*list1->maxm;// added to total revenue
corresponding to this action.
             }
             list1=list1->fptr;
             }
           }
```

```
        if(MAXIM>current->maxm)
         {
          current->maxm=MAXIM;
          current->max_ind=action;

              }
}
}//printing out current stage, movie playing, run length, rank of all
movies, and optimal action and optimal revenue.
if((stage==1)&&(mov==1))if(current->maxm > 0)
{
printf("\nstage %d movie %d run %d rank %d ranks",stage,mov,run,k);
for(j=1;j<=NUM_MOVIES/2;j++)if(con_set[stage][j]!=0)printf("%d",level
[l][j]);
printf("maxmoney %f optact %d",current->maxm,current->max_ind);
x=current->maxm;
for(j=1;j<=NUM_MOVIES/2;j++)if(con_set[stage][j]!=0)x=x*initial_pr[co
n_set[stage][j]][level[l][j]];
exp=exp+x;
fprintf(fout,"");
}
if(current->maxm>0)
{
current->fptr=(struct state *)malloc(sizeof(struct state));
current=current->fptr;
}
}

count=0;reduce=1;
for(j=1;j<=NUM_MOVIES/2;j++){if(con_set[stage][j]==0)count=count+1;}
for(j=1;j<=count;j++)reduce*=4;
l=l+reduce-1;
}

}
}

}
}

}
fprintf(fout,"EXPECTED REVENUE %f",exp);
return(0);
}//end of main;



//function for reading data from file


void readata()
{
      FILE *infile ;
      int j, k, l ;
      if((infile = fopen("datas18.dat","r")) == NULL )
          {
                    printf("could not open 'data1.dat'");
```

```
                    exit(1);
          };
/* the immediate rewards for any movie  */
      for(j = 1 ; j <= NUM_RANKS ; j++)
            {
                    for(k = 0 ; k < NUM_WEEKS ;k++)
                          {
                                fscanf(infile,"%f",&rev[j][k]);
                                printf("j %d k %d %f ",j,k,rev[j][k]);
                          }
                    printf("\n");
            }
/* assigning initial probabilities   */
      for(j = 1 ; j <= NUM_MOVIES ; j++)
       for(k = 1 ; k <= NUM_RANKS ;k++)
              {
                    fscanf(infile,"%f",&initial_pr[j][k]);
              }


/* assigning transition probability   */

      for(j =  1; j <= NUM_MOVIES ;j++)
         for(k = 1 ; k <= NUM_RANKS;k++)
              for(l = 1; l <= NUM_RANKS ; l++)
                   fscanf(infile,"%f",&ind_pr[j][k][l]);

/* release dates */
      for(j = 1 ; j <= NUM_MOVIES ; j++)
            {
                    fscanf(infile,"%d",&rel_dt[j]);
            }/*printf(" rel dates %d,%d",j,rel_dt[j]);}   */

/* obligation periods */
      for(k = 1 ; k <= NUM_MOVIES ;k++)
            {
                    fscanf(infile,"%d",&OPD[k]);
                    printf("opd,%d\n",OPD[k]);
            };

/* TYPE OF MOVIE*/
for(k=1;k<= NUM_MOVIES; k++)
      {
              fscanf(infile,"%d",&TYPE[k]);
      }
      fclose(infile);
}

///function to return probability of transition between states.
float get_prb(int l,int levl,int stage,int action, int runmv,int
act,int mov1)

{
  float prob=1;
  int mov,prev_rank,new_rank,pos,flag2;
  int i, j, flag;
```

```
  for(i=1;i<=NUM_MOVIES/2;i++)
       {
       mov=con_set[stage][i];// using "stage" variable to get
consideration set, and going movie by movie.
       if(mov==0)continue;
       prev_rank=level[1][i];// getting previous rank of movie using
1.
      if((mov!=mov1)&&(mov==action)&&(prev_rank==3))prob=0;
       pos=0;
       flag2=0;
       if  (mov==mov1)flag2=1;
       for(j=1;j<=NUM_MOVIES/2;j++)
       {
       if(con_set[stage+1][j]==mov)
       {//finding position of movie in next stage's consideration set.
       pos=j;
       break;
       }
       }
       if(pos!=0)
       {
       new_rank=level[lev1][pos];/ getting new rank.
       if((flag2=1)&&(mov!=action)&&(new_rank!=3))prob=0;
//if movie replaced and rank still not 3, prob=0.
       else if  ((flag2=1)&&(mov!=action)&&(new_rank==3))prob=prob;
//if movieis replaced and rank =3, then prob=1.
       else prob*=ind_pr[mov][prev_rank][new_rank];
//else use of individual transition probability.
       }
       }
// this was for the current running movie and action, now for other
movies.
   for(i=1;i<=NUM_MOVIES/2;i++)
       {
               flag=1;
               mov=con_set[stage+1][i];//one movie at next stage is
taken
               if(mov==0)continue;
               prev_rank=level[1][i];
               pos=0;
               for(j=1;j<=NUM_MOVIES/2;j++)
                       {

                               if(con_set[stage][j]==mov)
                                       {
                                               flag=0;
                                               break;
                                       }
                       }//if movie newly entered consideration set, its
 initial probability is considered.
               if(flag==1)
                       {
                               new_rank=level[lev1][i];


                               prob*=initial_pr[mov][new_rank];
```

```
                    }
        }
      flag=1;

for(i=1;i<=NUM_MOVIES/2;i++){if(con_set[stage+1][i]==action)flag=0;}
      if(flag==1)prob*=ind_pr[action][level[1][act]][runmv];
      return(prob);

}

//function generating matrix of rank combinitions.
void mat_gen(void)

{

  int q=1;
  int  i, j ;
  int m0, m1, m2, m3 ,m4, m5, m6,m7;


      for(m0=1;m0<=NUM_RANKS;m0++)
             for(m1=1;m1<=NUM_RANKS;m1++)
                    for(m2=1;m2<=NUM_RANKS;m2++)
                          for(m3=1;m3<=NUM_RANKS;m3++)
                              for(m4=1;m4<=NUM_RANKS;m4++)
                                for(m5=1;m5<=NUM_RANKS;m5++)
                                      for(m6=1;m6<=NUM_RANKS;m6++)
                                         for(m7=1;m7<=NUM_RANKS;m7++)
                                      {
                                      level[q][1]=m0;
                                      level[q][2]=m1;
                                      level[q][3]=m2;
                                      level[q][4]=m3;
                                      level[q][5]=m4;
                                      level[q][6]=m5;
                                      level[q][7]=m6;
                                      level[q][8]=m7;
                                      q++;
                                }
 }


// function generating the consideration set at each stage using the
data of released movies, movie types since type 3and 4 movies do not
stay in consideration set after 4 weeks.
 void conset()

    {
      int i,j,k,goodmv,stage,flag,l,flag1;
      int topset[NUM_MOVIES+10];
      for(i=0;i<=NUM_MOVIES;i++)topset[i]=0;
      j=1;
      for(i=1;i<= NUM_MOVIES;i++)
      {
      if(TYPE[i]==1)
            {
            topset[j]=i;//storing alltype1 and 2 movies.
```

```
            j++;
            }
      }
      for(i=j;i<=NUM_MOVIES;i++)topset[i]=0;
      for(stage=NUM_WEEKS;stage>=1;stage--)//for each stage.
      {    k=1;
       for(i=NUM_MOVIES;i>=1;i--)
        {    flag1=1;
            for(j=1;j<=NUM_MOVIES;j++)
            {
            if(topset[j]==i)
            {
            flag1=0;
            break;
            }
            }
            if((flag1==0)&&(rel_dt[i]<=stage))// if movie is good and
released then added to consideration set
            {
            con_set[stage][k]=i;
            k++;
            }
        }// else writing movies in con set until stage/2 or 8 movies
are written.
      if(((stage>=NUM_WEEKS/2)&&(k<=NUM_MOVIES/2))
          ||((stage<NUM_WEEKS/2)&&(k<=stage)))
      if (stage<NUM_WEEKS/2)j=stage;else j=NUM_MOVIES/2;
      while(k<=j)
      {
      for(i=NUM_MOVIES;i>=1;i--)
      {    flag=1;
          for(l=1;l<=k;l++)
            {
            if((i==con_set[stage][l])||(rel_dt[i]>stage))flag=0;
            }
          if(flag==1)
            {
            con_set[stage][k]=i;
            k++;
            }
      }}}
      for(i=NUM_WEEKS;i>=1;i--)
      {    printf("\nstage %d",i);
             for(j=1;j<=NUM_MOVIES/2;j++)
                    {printf("\t%d",con_set[i][j]);}
      }
    getchar();

 }
```

HEADER FILE FOR MDP MODEL

```
#ifndef PRMTRS
#define PRMTRS
#define NUM_WEEKS 8
#define NUM_MOVIES 16
#define NUM_LEVELS 6561 /* from combination of no. of ranks and no.
of movies*/
#define NUM_RANKS 3

int     con_set[10][NUM_MOVIES+10];
float   rev[NUM_RANKS+10][NUM_WEEKS+10]; /* revenues in a state and a
week */
float   initial_pr[NUM_MOVIES+10][NUM_RANKS+10];/* initial prob. */
float   ind_pr[NUM_MOVIES+10][NUM_RANKS+10][NUM_RANKS+10];/*
individual probabilities */
int     rel_dt[NUM_MOVIES+10]; /* release dates */
int     level[NUM_LEVELS + 100][NUM_MOVIES + 10];/* matrix to be
generated*/
int     OPD[NUM_MOVIES+10] ; /* obligation periods */
int     TYPE[NUM_MOVIES+10];
#endif
```

APPENDIX – 2

GA-CODE IN C++ LANGAUGE

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#define NUM_MOVIES 38
#define NUM_WEEKS 8
#define NUM_SCR 6
#define MAXSLOT 48
#include<graphics.h>
#include<conio.h>
#include<iomanip.h>
#include<string.h>
#include<fstream.h>
//#define NULL 0
//defining chromosome as a structure with b as an array of schedule
struct chrom
	{
		int no;
		int b[NUM_WEEKS][NUM_SCR];
		float revenue;
		int copies;
		struct chrom *next;
	};
void selection();
void crossover();
void mutation();
float evaluation(int b[NUM_WEEKS][NUM_SCR]);
int
rel_dt[NUM_MOVIES],demand[14][NUM_WEEKS+10],movie_type[NUM_MOVIES+10]
;
float contract_term[14][NUM_WEEKS+10];
int rrr,screen_cap[NUM_SCR];
float a[NUM_MOVIES][NUM_WEEKS][NUM_SCR];
struct chrom *head,*h;
int popsize,maxgen;
float pc,pm,tot_pop0,tot_pop1,tot_pc0,tot_pc1,tot_pm0,tot_pm1;
int gen=0;
main()
{
 FILE *fp;
 int ab,bc,cd,de,hpop=0,best_pop;
 float highest=0.0,hpc=0,hpm=0,tot_pop,tot_pc,tot_pm,best_pc,best_pm;
 static int best_revenue[3][3];
 int gdriver=DETECT,gmode=0,x,y,z,k,xx,count;
 int n,l,flag,flag1,flag2,r,m;
 struct chrom  *p,*q,*best;
 fp=fopen("scen16.dat","r");
 initgraph(&gdriver,&gmode,"");

 //giving screen capacities
```

```
screen_cap[0]=3472;screen_cap[1]=2736;screen_cap[2]=1728;screen_cap[3
]=1208; screen_ccap[4] 1112;screen_cap[5]=904;
contract_term[1][1] 0.1;contract_term[1][2]=0.3;contract_term[2][1]=0
.1;
contract_term[2][2] 0.2;contract_term[3][1]=0.2;contract_term[3][2]=0
.4; contract_term[4][1]=0.2;contract_term[4][2]=0.4;
 for(x=1;x<5;x++)
 for(y=3;y<=NUM_WEEKS;y++)
 {
      if(x==1)contract_term[x][y]=0.5;
      else if(x==2)contract_term[x][y]=0.3;
      else contract_term[x][y]=0.5;
 }

 for (x=0;x<NUM_MOVIES;x++)
      {
             fscanf(fp,"%d",&movie_type[x]); printf("
%d",movie_type[x]);
      }
      printf("\n");getch();

//writing demand for type of movies in each week
demand[1][0] 3675;demand[1][1]=2189;demand[1][2]=1306;demand[1][3]=78
0;
demand[1][4]=465;demand[1][5]=277;demand[1][6]=165;demand[1][7]=99;
demand[2][0] 4102;demand[2][1]=3280;demand[2][2]=2622;demand[2][3]=20
97;
demand[2][4]=1677;demand[2][5]=1341;demand[2][6]=1072;demand[2][7]=85
7;
demand[3][0]=1804;demand[3][1]=1157;demand[3][2]=742;demand[3][3]=476
; demand[3][4]=305;demand[3][5]=196;demand[3][6]=126;demand[3][7]=80;
demand[4][0]=1577;demand[4][1]=1358;demand[4][2]=1169;demand[4][3]=10
07;
demand[4][4]=867;demand[4][5]=747;demand[4][6]=643;demand[4][7]=554;
for(x=0;x<NUM_MOVIES;x++){
                               fscanf( fp," %d",&rel_dt[x]);
                               printf("%d ",rel_dt[x]);
                            }getch();
      for(ab=0;ab<2;ab++)
      for(bc=0;bc<2;bc++)
      for(cd=0;cd<2;cd++)
      for(de=0;de<4;de++)    //for loops for different settings of
parameters
      {
      gen=0;
      head= (struct chrom *)malloc(sizeof(struct chrom));
      h= head; clrscr();
            switch(ab)
                  {
                        case 0:popsize=300;break;
                        case 1:popsize=400;break;
                  }
            switch(bc)
                  {
                        case 0:pc=0.95;break;
                        case 1:pc=0.99;break;
```

```
                }
            switch(cd)
                {
                        case 0:pm=0.03;break;
                        case 1:pm=0.05;break;
                }
            switch(de)
                {
                        case 0:rrr=1;break;
                        case 1:rrr=25;break;
                        case 2:rrr=100;break;
                        case 3:rrr=400;break;
                        case 4:rrr=6000;break;
                }

      maxgen=2000;
      pc*=100;
      gotoxy(10,10);printf("give random seed  ");//getting random
seed for random number generator.
      scanf("%d",&rrr);clrscr();
      srand(rrr);
      pm*=100;
//creating initial population
 for(x=0;x<popsize;x++)
  {
   h->next=(struct chrom *)malloc(sizeof(struct chrom));
   if(h->next==NULL){printf("could not assign");exit(1);}
   h->next->no=x;h->next->copies=0;h->next->next=NULL;
   for(y=0;y<NUM_WEEKS;y++)
   for(z=0;z<NUM_SCR;z++)
       {    flag=0;
            while(flag==0)
              {
                flag1=0;
                r=rand()% NUM_MOVIES;
                for(l=0;l<z;l++)if(r==h->next->b[y][l])flag1=1;
   //considers release date for initial population generation.
               if((rel_dt[r]<=y)&&(flag1==0))
                {
                     h->next->b[y][z]=r;
                     flag=1;
                }
              }
       }
 ///using repairing function to make chromosomes feasible.
       for(xx=0;xx<NUM_SCR-2;xx++)
       for(y=2;y<NUM_WEEKS;y++)
       for(z=0;z<NUM_SCR;z++)
         {    flag1=1;
             r=h->next->b[y][z];
             flag=1;
             for(k=0;k<NUM_SCR;k++)
             if(r==h->next->b[y-1][k]){flag=0;flag1=0;}
             if(flag==1)
               {
                 for(m=0;m<y-1;m++)
```

```
              for(k=0;k<NUM_SCR;k++)

              if(r==h->next->b[m][k]){flag=0;break;}
            }
// checking for discontinuity of movie in b[y][z];

        if((flag==0)&&(flag1==1))  //movie discontinuous
          {         count=0;
            for(m=0;m<y-1;m++)
            for(k=0;k<NUM_SCR;k++)
            if(r==h->next->b[m][k]){count++;}
            k=0;
            flag2=0;
            if(count>NUM_WEEKS/2)
            while(flag2==0)
              {
               flag1=0;
               n=h->next->b[y-1][k];
               for(l=0;l<NUM_SCR;l++)
               if(n==h->next->b[y][l])flag1=1;
               if(flag1==0)
               {
                 h->next->b[y-1][k]=h->next->b[y][z];
                 flag2=1;
        }
        k++;// scheduling movie in the weeks where it was not scheduled
 removing discontinuity.
 }
 else
 while(flag2==0)
     {  //removing the movie, because it was for less number of times.
        flag1=0;
        n=h->next->b[y-1][k];
        for(l=0;l<NUM_SCR;l++)
        if(n==h->next->b[y][l])flag1=1;
        if(flag1==0)
           {
             h->next->b[y][z]=n;
             flag2=1;
           }
        k++;
     }
 }
 }
 h->next->revenue=evaluation(h->next->b); //evaluating chromosome.
 h=h->next;
 }
 h=head;
 h=head;best=((struct chrom *)malloc(sizeof(struct chrom)));
 best->revenue=h->next->revenue;
 for(y=0;y<NUM_WEEKS;y++)
 for(z=0;z<NUM_SCR;z++)
     best->b[y][z]=h->next->b[y][z];// finding the best schedule
 for(x=0;x<popsize;x++)
       {
       if(h->next->revenue>best->revenue)
             {
```

```
                best->revenue=h->next->revenue;
                for(y=0;y<NUM_WEEKS;y++)
                for(z=0;z<NUM_SCR;z++)
                best->b[y][z]=h->next->b[y][z];
            }
                h h->next;
            } gotoxy(10,10);
printf("\nbest revenue = %f",best->revenue);
h=head; clrscr();
gotoxy(20,20);
printf("Please wait till I display the result");
for(k=0;k<maxgen;k++)
        { //moving generation to generation
        srand(rrr);
        selection();
        srand(rrr);
        crossover();
        srand(rrr);
        mutation();
        h-head;
        p-head >next;q-head->next;
        for(x-0;x<popsize;x++)
            {
                if(h->next->revenue>p->revenue){p=h-
>next;}//finding best
                if(h->next->revenue<q->revenue){q=h-
>next;}//finding worst
                        h=h->next;
            }
        if(best->revenue>q->revenue)
        {
        q->revenue=best->revenue;//replacing worst by previous best.
        for(y=0;y<NUM_WEEKS;y++)
         for(z=0;z<NUM_SCR;z++)
           q->b[y][z]=best->b[y][z];
          if(best->revenue>p->revenue)
            {
            p->revenue=best->revenue;
            for(y-0;y<NUM_WEEKS;y++)
            for(z=0;z<NUM_SCR;z++)
            p->b[y][z]=best->b[y][z];
            }
         else {  //replacing previous best for current best
             best->revenue=p->revenue;
             for(y=0;y<NUM_WEEKS;y++)
             for(z=0;z<NUM_SCR;z++)
             best->b[y][z]=p->b[y][z];
            }

        }
        if(((k+1)%100==0)||(k==maxgen-1))
            {
             clrscr();gotoxy(15,15);
             printf("%d     %f    %f",popsize, pc,pm);gotoxy(15,15);
             printf("best revenue = %f , generation %d",best-
>revenue,k+1);
            }
```

```
        if((k==maxgen-1))
              {
                printf("\n");
                for(y=0;y<NUM_WEEKS;y++)
                   {for(z=0;z<NUM_SCR;z++)
                       printf("    %d", best->b[y][z]);
                       printf("\n");
                   }
              }
         gen++;
         if(best->revenue>highest)
            {
              highest=best-
>revenue;hpop=popsize;hpc=pc;hpm=pm;//changing highest for current
setting.
            }
        }
        best_revenue[ab][bc]+=best->revenue;
         p=head;q=head;
         while(q!=NULL)
              {
                     p=q->next;
                     free(q);
                     q=p;
              }
              free(best);
         }
         tot_pop0=0;tot_pop1=0;tot_pc0=0;tot_pc1=0;tot_pm0=0;tot_pm1=0;
         for(bc=0;bc<2;bc++)
         for(cd=0;cd<2;cd++)
              {
                     tot_pop0+=best_revenue[0][bc];
                     tot_pop1+=best_revenue[1][bc];
              }
         for(ab=0;ab<2;ab++)
         for(cd=0;cd<2;cd++)
              {
                      tot_pc0+=best_revenue[ab][0];
                      tot_pc1+=best_revenue[ab][1];
              }
         for(ab=0;ab<2;ab++)
         for(bc=0;bc<2;bc++)
              {
                     tot_pm0+=best_revenue[ab][bc][0];
                     tot_pm1+=best_revenue[ab][bc][1];
              }
         if(tot_pop0>tot_pop1)best_pop=300;else best_pop=400;
         if(tot_pc0>tot_pc1)best_pc=0.95;else best_pc=0.99;
         if(tot_pm0>tot_pm1)best_pm=0.03;else best_pm=0.05;
         printf("\n");
         printf(" best popsize=%d, bestpc=%f,
bestpm=%f",best_pop,best_pc,best_pm);
         printf("\nhighest rev=%f,hpop=%d,hpc=%f,hpm=%f",highest,
hpop,hpc,hpm);

              closegraph();
         getch();
```

```
        return(0);
 }


void selection()//selection process.
{
int sel[MAXSLOT+1500],flag,selection[MAXSLOT+1500];
int x,y,z,r,n,k;
struct chrom *p,*q;
p=head->next;
for(n=0;n<2;n++)
   {
    z=popsize;p=head->next;
    x=0;
    while(p!=NULL)
       {
              p->no=x;
              x++;
              p=p->next;
       }
       for(x=0;x<popsize;x++)sel[x]=x;
       for(x=0;x<popsize-2;x++)
              {
                     r=rand()% z;
                     selection[x]=sel[r];//chromosomes selected randomly
                     for(y=r;y<z;y++)sel[y]=sel[y+1];
                     z-=1;
              }
       selection[x]=sel[0];
       selection[x+1]=sel[1];
       for(x=0;x<popsize-1;x++)
              {      y=selection[x];
                       p=head;q=head;
                       while(p->next->no!=y){p=p->next;}
                       z=selection[x+1];
                       while(q->next->no!=z){q=q->next;}
// comparing chromosomes and increasing the copies of winner.
                       if(p->next->revenue>q->next->revenue)p->next-
>copies++;
                       else q->next->copies++;
                       x++;
              }
       }
 p=head;x=0;
 q=head->next;
 p=head;
 while((q!=NULL)&&(q->no<popsize))
        {
              if(q->copies==0)
                     {  //removing  chromosomes with 0 copies for next
 generation
                            p->next=q->next;
                            free(q);
                            q=p->next;
                     }
              else
                     {
```

```
                            p=q;
                            q=p->next;
                      }
          }
p=head;
while(p->next!=NULL)
       {     //making copies for chromosomes
              z=p->next->copies;
              if(z!=0)
              for(x=2;x<=z;x++)
                    {
                            q=(struct chrom *)(malloc(sizeof(struct
chrom)));
                            q->revenue=p->next->revenue;
                            q->copies=0;
                            for(y=0;y<NUM_WEEKS;y++)
                                   for(k=0;k<NUM_SCR;k++)
                                          {q->b[y][k]=p->next->b[y][k];}
                            q->next=p->next->next;
                            p->next->next=q;
                    }
              p=p->next;
       }
p=head; x=0;
while(p->next!=NULL)
       {
              p->next->no=x;
              p->next->copies=0;
              p=p->next;
              x++;
       }
p=head;

}

//evaluation function evaluates revenue corresponding to each
schedule
float evaluation(int b[NUM_WEEKS][NUM_SCR])
{
 int x,y,z,k,xx,yy,run_length,temp,xxx;
 float revenue=0.0;
 for(x=0;x<NUM_WEEKS;x++)
       for(y=0;y<NUM_SCR;y++)
       {        run_length=0;
              k=b[x][y]; // finding movie which is scheduled in b[x][y]
              for(xx=0;xx<x;xx++)
              for(yy=0;yy<y;yy++)
              if(b[xx][yy]==k)run_length++;// calculating run length
 for movie
              run_length++;if(k<6)run_length++;
// for first six movies they are one week older than the first week
so their runlength is increased by 1.
              xxx=x-rel_dt[k];
//comparing demand of movie with screen capacity.
              if(demand[movie_type[k]][xxx] > screen_cap[y])
              temp= screen_cap[y];
              else
```

```
              temp=demand[movie_type[k]][xxx];
//calculating revenue.
              revenue+=temp*contract_term[movie_type[k]][run_length];
       }
 return(revenue);
 }
```

```
// crossover function first selects randomly two chromosomes each
time, generates random number; if less than pc, selects crossover
site randomly and generates two children.
 void crossover()
 {
      int x,y,xx,z,k,r,r1,r2,l,temp[NUM_WEEKS+10][NUM_SCR+10],count;
      struct chrom *p,*q;
      int cross[MAXSLOT+1500],flag,flag1,flag2,m,n;
      int crosspair[MAXSLOT+1500];
      z=popsize;p=head->next;x=0;
      while(p!=NULL)
              {
              x++;
              p=p->next;
              }
      for(x=0;x<popsize;x++)cross[x]=x;
      for(x=0;x<popsize-2;x++)
              {
                     r=rand()%z;
                     crosspair[x]=cross[r];
                     for(y=r;y<z;y++)cross[y]=cross[y+1];
                     z-=1;
              }
      crosspair[x]=cross[0];
      crosspair[x+1]=cross[1];
      for(x=0;x<popsize-1;x++)
              {       y=crosspair[x];
                      p=head;q=head;
                      while(p->next->no!=y){p=p->next;}//selected first
parent
                      z=crosspair[x+1];
                      while(q->next->no!=z){q=q->next;}//selected second
parent
                      r= rand()% 101;//generated random number
if(r<pc)
 {
    r1=rand()% MAXSLOT;// random number for crossover site.
    r2=(r1%NUM_SCR);
// does crossover between two parents
    for(k=r2;k<NUM_WEEKS;k++)
    for(l=0;l<NUM_SCR;l++)
         {
            temp[k][l]=p->next->b[k][l];
         }
    for(k=r2;k<NUM_WEEKS;k++)
    for(l=0;l<NUM_SCR;l++)
       {
        p->next->b[k][l]=q->next->b[k][l];
       }
```

```
      for(k=2;k<NUM_WEEKS;k++)
      for(l=0;l<NUM_SCR;l++)
         {
          q->next->b[k][l]=temp[k][l];
         }
  }

  x++;
}
p=head;
while(p->next!=NULL)
{  // uses same correction function as that of initial population.
   for(xx=0;xx<NUM_SCR-2;xx++)
   for(y=2;y<NUM_WEEKS;y++)
   for(z=0;z<NUM_SCR;z++)
      {
        flag1=1;
        r=p->next->b[y][z];
        flag=1;
        for(k=0;k<NUM_SCR;k++)
        if(r==p->next->b[y-1][k]){flag=0;flag1=0;}
        if(flag==1)
              {
                for(m=0;m<y-1;m++)
                for(k=0;k<NUM_SCR;k++)
                if(r==p->next->b[m][k]){flag=0;break;}
              }
        if((flag==0)&&(flag1==1))
              {
               count=0;
               for(m=0;m<NUM_WEEKS;m++)
               for(k=0;k<NUM_SCR;k++)
               if(r==p->next->b[m][k]){count++;}
               k=0;
               flag2=0;
               if(count<NUM_WEEKS/2)
                  while(flag2==0)
                     {
                       flag1=0;
                       n=p->next->b[y-1][k];
                       for(l=0;l<NUM_SCR;l++)
                       if(n==p->next->b[y][l])flag1=1;
                       if(flag1==0)
                        {
                          p->next->b[y-1][k]=p->next->b[y][z];
                          flag2=1;
                        }
              k++;
        }
        else
        while(flag2==0)
         {
           flag1=0;
           n=p->next->b[y-1][k];
           for(l=0;l<NUM_SCR;l++)
              if(n==p->next->b[y][l])flag1=1;
              if(flag1==0)
```

```
                {
                   p->next->b[y][z]=n;
                   flag2=1;
                }
            k++;
        }
}
}
p=p->next;
}
p=head;
while(p->next!=NULL)
  {
      p->next->revenue=evaluation(p->next->b);
      p=p->next;
  }
}
```

```
//mutation function, for each bit, generates random number, checks
if<pm. if yes, replaces the movie at that bit with movie which was
not at all scheduled. If no such movie present then selects a movie
which is released be not played on some other screen in the same
week. If random no. is <pm/2, screens between movies of same week are
exchanged.
void mutation()
 {
       struct chrom *p, *q;
       int
x,y,z,k,r,m,n,l,flag,flag1,flag2,xx,yy,ch_mov,rr,temp1,xxx,count;
       p=head;
       while(p->next!=NULL)
       {
          for(xx=0;xx<NUM_WEEKS;xx++)
           {
           for(yy=0;yy<NUM_SCR;yy++)
            {
              r=rand()%101;//random no.
              if(r<pm)
                {
                 for(x=0;x<NUM_MOVIES;x++)
                     {
                       flag=0;
                       for(y=0;y<NUM_WEEKS;y++)
                       for(z=0;z<NUM_SCR;z++)
                            {
                            if(x==p->next->b[y][z])
                                 {
                                         flag=1;
                                 }
                            }
 if((flag==0)&&(rel_dt[x]<=xx)){ch_mov=x;break;}
 }
 if(flag==0)p->next->b[xx][yy]=ch_mov;
```

```
// replaced by movie which was not at all scheduled.
else
{//replaces by movie which was not scheduled in this week.
  for(x=0;x<NUM_MOVIES;x++)
        {
            flag=0;
            for(z=0;z<NUM_SCR;z++)
                {
                if(x==p->next->b[xx][z])
                        {
                        flag=1;
                        }
                }
          if((flag==0)&&(rel_dt[x]<=xx)){p->next->b[xx][yy]=x; break;}
         }
}
if(r<pm/2)
{   // exchanging screens.
    rr=rand()%NUM_SCR;
    temp1= p->next->b[xx][rr];
    if((temp1>NUM_MOVIES)||(temp1<0))
        {
         printf("oh no");
         getch();
        }
    p->next->b[xx][rr]=p->next->b[xx][yy];
    if((p->next->b[xx][rr]>NUM_MOVIES)||(p->next->b[xx][rr]<0))
        {
         printf("oh no1");
         getch();
        }
    p->next->b[xx][yy]=temp1;
    if((p->next->b[xx][yy]>NUM_MOVIES)||(p->next->b[xx][yy]<0))
        {
         printf("oh no2");
         getch();
        }
 }
 if((gen==maxgen-1))
  {
   printf("\n");
   for(y=0;y<NUM_WEEKS;y++)
      {
       for(z=0;z<NUM_SCR;z++)
       if((p->next->b[y][z]>NUM_MOVIES)||(p->next->b[y][z]<0))
          {
           printf("\ni am here%d no. %d",p->next->b[y][z],p->next->no);
           getch();
          }
      }
   }

 }
 }
 }
 //using same repairing function
 for(xxx=0;xxx<NUM_SCR-2;xxx++)
```

```
for(y=2;y<NUM_WEEKS;y++)
for(z=0;z<NUM_SCR;z++)
  {
   flag1=1;
   r=p->next->b[y][z];
   flag=1;
   for(k=0;k<NUM_SCR;k++)
   if(r==p->next->b[y-1][k]){flag=0;flag1=0;}
   if(flag==1)
      {
       for(m=0;m<y-1;m++)
       for(k=0;k<NUM_SCR;k++)
       if(r==p->next->b[m][k]){flag=0;break;}
      }

   if((flag==0)&&(flag1==1))
      {
       count=0;
       for(m=0;m<y-1;m++)
       for(k=0;k<NUM_SCR;k++)
       if(r==p->next->b[m][k]){count++;}
       k=0;
       flag2=0;
       if(count>NUM_WEEKS/2)
       while(flag2==0)
         {
           flag1=0;
           n=p->next->b[y-1][k];
           for(l=0;l<NUM_SCR;l++)
           if(n==p->next->b[y][l])flag1=1;
           if(flag1==0)
               {
                p->next->b[y-1][k]=p->next->b[y][z];
                flag2=1;
               }
           k++;
         }
        else
        while(flag2==0)
          {
           flag1=0;
           n=p->next->b[y-1][k];
           for(l=0;l<NUM_SCR;l++)
           if(n==p->next->b[y][l])flag1=1;
           if(flag1==0)
           {
              p->next->b[y][z]=n;
              flag2=1;
           }
           k++;
          }
       }
 }
 if((gen==maxgen-1))
    {printf("\n");
   for(y=0;y<NUM_WEEKS;y++)
       {for(z=0;z<NUM_SCR;z++)
```

```
        if((p->next->b[y][z]>NUM_MOVIES)||(p->next->b[y][z]<0))
          {
            printf("\n%d no. %d",p->next->b[y][z],p->next->no);
            getch();
          }
         }
      }
p=p->next;
}
p=head;
while(p->next!=NULL)
        {
          p->next->revenue=evaluation(p->next->b);p=p->next;
        }
}
```

Sample data file giving the type and release date for movies:

```
2 1 1 1 3 1 1 1 4 3 3 1 3 4 2 1 3 4 1 4 3 1 3 3 1 2 4 1 4 3 1 3 1 3 2
3 1 3
 -1 -1 -1 -1 -1 -1 0 0 0 0 1 1 1 1 2 2 2 2 3 3 3 3 4 4 4 4 5 5 5 5 6
6 6 6 7 7 7 7
```

References

Abad, P.L. (1996), "Optimal Pricing and Lot-Sizing under Conditions of Perishability and Partial Backordering," *Management Science*, 42 (8),1093-1104.

Abraham, Magid M. and Leonard M. Lodish (1993), "An Implemented System for Improving Promotion Productivity Using Store Scanner Data," *Marketing Science*, 12(3), 248--269.

Acquilano, N. J., and R. B. Chase (1991), *Fundamentals of Operations Management*, Irwin, Homewood, IL.

Ahuja, Ravindra K., Thomas L. Magnanti, and James B. Orlin (1993), *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, Englewood Cliffs, NJ.

Austin, Bruce A. and Thomas F. Gordon (1987), "Movie Genres: Toward a Conceptualized Modeland Standardized Definition," in *Current Research in Film: Audiences, Economics and the Law, Vol. 4*, B. A. Austin, ed., Norwood, NJ: Ablex Publishing Co.

Bagchi, T.P. (1999), " Multiobjective Scheduling by Genetic Algorithms," Kluwer Academic Publishers, Dordrecht, The Netherlands.

Bagchi, T.P., K. Deb, "Calibration of G.A. Parameters: The Design of Experiments Approach," *Computer Science & Informatics*, 26, 3.

Balakrishnan, P.V. (Sundar) and Jacob, V.S. (1996), "Genetic Algorithms for Product Design," *Management Science*, 42 (August), 1105-1117.

Baker, Kenneth R. (1993), *Elements of Sequencing and Scheduling*, Dartmouth College, Hanover, NK.

Bellman, Richard E. (1957), *Dynamic Programming*, Princeton University Press, Princeton, NJ.

Bertsekas, Dimitri P. (1987), *Dynamic Programming: Deterministic and Stochastic Models*, Prentice Hall, Englewood Cliffs, NJ.

Blattberg, R. C. and S. J. Hoch (1990), "Database Models and Managerial Intuition: 50% Model + 50% Manager," *Management Science*, 36, 8 (August), 887--899.

Borin, Norm, Paul W. Farris, and James R. Freeland (1994), "A Model for Determining Retail Product Category Assortment and Shelf Space Allocation," *Decision Sciences*, 25 (3), 359--384.

Bultez, Alain and Phillippe Naert (1988), "S.H.A.R.P.: Shelf Allocation for Retailers' Profit," *Marketing Science*, 7, 3 (Summer), 211--231.

Chakraborti, S., Sudipta De, K. Deb (1999), "Model-based Object Recognition from a Complex Binary Imagery using Genetic Algorithms," EvolASP.

Corstjens, M. and Doyle, P. (1981), " A Model for Optimizing Retail Space Allocations," *Management Science*, 27 (July), 822-833.

Derman, C. (1963), "On Optimal Replacement Rules When Changes of State are Markovian," in *Mathematical Optimization Techniques*, R. Bellman (ed.), University of California Press, Berkeley, California, 201-210.

De Vany, Arthur S. and W. David Walls (1997), "The Market for Motion Pictures: Rank, Revenue, and Survival," *Economic Inquiry*, 35 (October), 783-797.

Dodds, John C. and Morris B. Holbrook (1988), "What's an Oscar Worth? An Empirical Estimation of the Effect of Nominations and Awards on Movie Distribution and Revenues," *Current Research in Film: Audiences, Economics and the Law, Vol. 4*, B. A. Austin, ed., Norwood, NJ: Ablex Publishing Co.

Eliashberg, Jehoshua; Jonker, Jedid-Jah; Sawhney, Mohanbir S.; and Wierenga, Berend (2000), "MOVIEMOD: An implementable decision support system for pre-release market evaluation of motion pictures" *Marketing Science*, Volume 19, Number. 3, pp. 226-243.

Eliashberg, Jehoshua and Mohanbir S. Sawhney (1994), "Modeling Goes to Hollywood: Predicting Individual Differences in Movie Enjoyment," *Management Science*, 40 (September),1151-1173.

Eliashberg, Jehoshua and Steven M. Shugan (1997), "Film Critics: Influencers or Predictors?"*Journal of Marketing*, 61, 2 (April), 68-78.

Eliashberg , Jehoshua, Sanjeev Swami, Charles B. Weinberg and Berend Wierenga (2001), "Implementing and Evaluating SILVERSCREENER: A Marketing Management Support System for Movie Exhibitors," forthcoming *Interfaces*: Special issue on Marketing Engineering.

Goldberg, D.E. (1989), " Genetic Algorithms in Search, Optimization, and Machine Learning," *Addison-Wesley,Reading*, MA.

Fourer, Robert, David M. Gay, and Brian W. Kernighan (1993), *AMPL: A Modeling Language for Mathematical Programming*, The Scientific Press, San Francisco, CA.

Grefenstette, J. J.(1986), "Optimization of Control Parameters for Genetic Algorithms," *IEEE Trans. Systems, Man and Cybernetics*, SMC 16, 1, 122-128.

Hirschman, Elizabeth C. and Morris B. Holbrook (1982), "Hedonic Consumption: Emerging Concepts, Methods and Propositions,"*Journal of Marketing*, 46 (Summer), 92-101.

Holbrook, Morris B. and Elizabeth C. Hirschman (1982), "The Experiential Aspects of Consumption: Consumer Fantasies, Feelings and Fun," *Journal of Consumer Research*, 9 (September), 132-140.

Holland, J. H.(1975), "Adaptation in Natural and Artificial Systems," *The University of Machigan Press*, Ann Arbor, MI.

Hoch, S. J. (1994), "Experts and Models in Combination,"*The Marketing Information Revolution*, R. C. Blattberg, R. Glazer, and J. D. C. Little (eds.), Harvard Business School Press, Boston, MA.

Jain, Karuna and Edward A. Silver (1994), "Lot Sizing for a Product Subject to Obsolescence or Perishability," *European Journal of Operational Research*, 75 (2), 287--295.

Jedidi, Kamel, Robert E. Krider, and Charles B. Weinberg (1998), "Clustering at the Movies," *Marketing Letters*, 9 (4), 393-405.

Jones, Morgan and Christopher J. Ritz (1991), "Incorporating Distribution into New Product Diffusion Models," *International Journal of Research in Marketing*, 8, 91-112.

Joshi P. (2000), "Performance of Genetic Algorithm and Greedy Heuristic with respect to Distribution of Local Solutions on Hypercubes," Thesis Report, M.Tech., Indian Institute of Technology, Kanpur, India.

Karmarkar, Uday S. (1996),"Integrative Research in Marketing and Operations Management," *Journal of Marketing Research*, 33 (May), 125-133.

Krider, Robert E. and Charles B. Weinberg (1998), "Competitive Dynamics and the Introduction of New Products: The Motion Picture Timing Game," *Journal of Marketing Research*, 35, 1 (February), 1-15.

Kumar S., Bagchi, T.P., and Sriskandarajah (2000), "Lot streaming and scheduling heuristics for *m*-machine no-wait flowshops," *Computers and Industrial Engineering*, 38, 149-172.

Lehmann, Donald R. and Charles B. Weinberg (2000), "Sales Via Sequential Distribution Channels: An Application to Movie Audiences,"*Journal of Marketing*, 64 (3), 13-33.

Liepins, G.E. and W.D. Potter (1991), "A Genetic Algorithm Approach to Multiple Fault Diagnosis," *Handbook of Genetic Algorithms,* L. Davis, (Ed.) VanNostrand Reinhold, New York.

Littlewood, K. (1972), "Forecasting and Control of Passenger Bookings," *AGIFORS 12th Annual Symposium Proceedings*, 95--128.

Lodish, Leonard M. (1971), "CALLPLAN: An Interactive Salesman's Call Planning System," *Management Science*, 18, 4(2), 25--40.

Mahajan, Vijay, Eitan Muller, and Roger Kerin (1984), "Introduction Strategy for New Products with Positive and Negative Word-of-Mouth," *Management Science*, 30 (December), 1389-1404.

Michalewicz, Z.(1992), "Genetic Algorithms + Data Structures = Evolution Programs," *Springer-Verlag*, Berlin.

Murata, T. and Ishibuchi,(1994) "Performance evaluation of genetic algorithms for flowshop scheduling problems," *Proceedings of the IEEE Conference on Genetic Algorithms,* 1994, 812-817.

Prasad, Ashutosh, Vijay Mahajan, and Bart J. Bronnenberg (1998), "Product Entry Timing in Dual Distribution Channels: The Case of the Movie Industry," Working Paper, University of Texas at Austin.

Puterman, Martin L. (1994), *Markov Decision Processes: Discrete Stochastic Dynamic Programming,* John Wiley & Sons, New York, NY.

Radas, Sonja and Steven M. Shugan (1998), "Seasonal Marketing and Timing Introductions," *Journal of Marketing Research*, 35 (August), 296-315.

Rangaswamy, Arvind (1993), "Marketing Decision Models: From Linear Programs to Knowledge-based Systems," *Handbooks in OR & MS*, J. Eliashberg and G. L. Lilien (eds.), Elsevier Science Publishers B.V., The Netherlands, 733--771.

Reddy, Srinivas K., Jay E. Aronson, and Antonie Stam (1998), "SPOT: Scheduling Programs Optimally on Television," *Management Science*, 44 (1), 83-102.

Saxena, Sudhir (2000), "Implimentation of a Marketing Decision Support System for Motion Picture Retailing," Thesis Report, M.Tech., Indian Institute of Technology, Kanpur, India.

Sawhney, Mohanbir S. and Jehoshua Eliashberg (1996), "A Parsimonious Model for Forecasting Gross Box Office Revenues of Motion Pictures," *Marketing Science*, 15 (2), 113-131.

Smith, Sharon, P. and V. Kerry Smith (1986), "Successful Movies: A Preliminary Empirical Analysis," *Applied Economics*, 18, 501-507.

Squire, Jason E. (1992), *The Movie Business Book*, 2nd Ed., New York: Simon & Schuster, Inc.

Swami, Sanjeev (1998), "*Dynamic Marketing Decisions in the Presence of Perishable Demand*," Ph.D. Dissertation, University of British Columbia, Vancouver, Canada.

Swami, Sanjeev, Eunkyu Lee, and Charles B. Weinberg (1998), "Optimal Channel Contracts for Marketing Perishable Products," Working Paper, University of British Columbia.

Swami, Sanjeev, Jehoshua Eliashberg, and Charles B. Weinberg (1999), "SilverScreener: A Modeling Approach to Movie Screens Management," *Marketing Science*, 18 (3), pp. 352-372.

Variety, The International Entertainment Weekly.

Vogel, Harold L. (1994), *Entertainment Industry Economics: A Guide for Financial Analysis*, 2nd ed., New York: Cambridge University Press.

Weinberg, Charles B. (1986), "ARTS PLAN: Implementation, Evolution and Usage," *Marketing Science*, 5 (2), 143--158.

Zufryden, Fred S. (1996), "Linking Advertising to Box Office Performance of New Film Releases-A Marketing Planning Model," *Journal of Advertising Research*, (July-August), 29-41.

A 133737

Date Slip

This book is to be returned on
1e date last stamped. t

....

....

....

....

....

....

....

....

....

....

....

....